2011-09

Web Applications and thin clients in the Navy

Britt, Jeremy L.

Monterey, California. Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

WEB APPLICATIONS AND THIN CLIENTS IN THE NAVY

by

Jeremy L. Britt

September 2011

Thesis Advisor:	Glenn Cook
Second Reader:	Doug MacKinnon

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** September 2011	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE Web Applications and Thin Clients in the Navy			**5. FUNDING NUMBERS**
6. AUTHOR(S) Jeremy L. Britt			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol number N.A.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

This thesis investigates the advantages and disadvantages of transitioning to Web Applications and Thin Client-Server Architecture for U.S. Navy shore based Components. Thin Clients and Web Technology have advanced significantly over the last few years and now more than ever, offer a multitude of cost efficient solutions. In the past, networking technology and bandwidth limitations made traditional Personal Computers or "Fat Clients" a more viable option for Naval Commands. The advancements in networking technology and Wi-Fi have significantly reduced these constraints. Moore's Law has held constant, advancing digital storage and processing capability far beyond the traditional Client-Server Architecture's ability to take full advantage of these services. The proliferation of server and network technology continues to provide economies of scale that drive down the cost of hardware. The accessibility of these technologies enabled application and software developers to steadily increase the size and complexity of software and applications. The Fat Client's proliferation led to most of this software and application development in the form of Native Applications. The cost of Software and Native Applications written for Fat Client platforms continues to increase while Server utilization remains negligible. Decentralization due to the inherent local access precipitated by the use of Fat Client-Server architectures and Native Applications creates surplus server capacity and redundant data centers. The Department Of The Navy's (DON) focus is shifting to Thin Clients and Enterprise Software Licensing due to budgetary constraints and the need for increased efficiencies. It is possible that Thin Client-Server Architecture and Web Applications may be able to provide these cost savings and efficiencies.

14. SUBJECT TERMS Web Applications, Native Applications, Mobile Applications, Client Server Architecture, Thin Client Server Architecture, Fat Client, Thin Client, Smart Mobile Wireless Devices, Total Cost of Ownership, Energy Consumption			**15. NUMBER OF PAGES** 109
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

WEB APPLICATIONS AND THIN CLIENTS IN THE NAVY

Jeremy L. Britt
Lieutenant, United States Navy
B.B.A., University of North Florida, 2007

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY MANAGEMENT

from the

NAVAL POSTGRADUATE SCHOOL
September 2011

Author: Jeremy L. Britt

Approved by: Glenn R. Cook
Thesis Advisor

Dr. Doug MacKinnon
Second Reader

Dr. Dan Boger
Chair, Department of Information Sciences

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

This thesis investigates the advantages and disadvantages of transitioning to Web Applications and Thin Client-Server Architecture for U.S. Navy shore based Components. Thin Clients and Web Technology have advanced significantly over the last few years and now more than ever, offer a multitude of cost efficient solutions. In the past, networking technology and bandwidth limitations made traditional Personal Computers or "Fat Clients" a more viable option for Naval Commands. The advancements in networking technology and Wi-Fi have significantly reduced these constraints. Moore's Law has held constant, advancing digital storage and processing capability far beyond the traditional Client-Server Architecture's ability to take full advantage of these services. The proliferation of server and network technology continues to provide economies of scale that drive down the cost of hardware. The accessibility of these technologies enabled application and software developers to steadily increase the size and complexity of software and applications. The Fat Client's proliferation led to most of this software and application development in the form of Native Applications. The cost of Software and Native Applications written for Fat Client platforms continues to increase while Server utilization remains negligible. Decentralization due to the inherent local access precipitated by the use of Fat Client-Server architectures and Native Applications creates surplus server capacity and redundant data centers. The Department Of The Navy's (DON) focus is shifting to Thin Clients and Enterprise Software Licensing due to budgetary constraints and the need for increased efficiencies. It is possible that Thin Client-Server Architecture and Web Applications may be able to provide these cost savings and efficiencies.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

AMPS	Advanced Mobile Phone Service
API	Application Programming Interfaces
ASP	Active Server Pages
CAC	Common Access Card
C-Cubed	Computer Center Corporation
CDMA	Code Division Multiple Access
CGI	Common Gateway Interface
CNO	Chief of Naval Operations
CNRFC	Commander Naval Reserve Forces Command
COBOL	Common Business Oriented Language
COTS	Commercial Off -the-Shelf
CSS	Cascading Style Sheet
DEC	Digital Equipment Corporation
DBMS	Database Management Server
DDCIO (N)	Department of the Navy Deputy Chief Information Officer
DoD-	Department of Defense
DON	Department of the Navy
EDVAC	Electronic Discrete Variable Computer
EJB	Enterprise JavaBeans Technology
ENIAC	Electronic Numerical Integrator and Computer
FDMA	Frequency Division Multiplexing Access
FORTRAN	Formula Translation
GSM	Global System for Mobile communication standard
GUI	Graphical User Interface
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IEEE	Institute of Electrical and Electronics Engineers
IT	Information Technology
ITU	International Telecommunications Union
JS	JavaScript

JSP	Java Server Pages
PC	Personal Computer
RAM	Random Access Memory
TCO	Total Cost of Ownership
TDMA	Time Division Multiple Access
UNIVAC	Universal Automatic Computer
U.S.	United States
Wi-Fi	Wireless Fidelity
1-G	First Generation
2-G	Second Generation
3-G	Third Generation

ACKNOWLEDGMENTS

The author would like to express his tremendous gratitude and appreciation to Professor Glenn Cook for his patience, guidance, and profound knowledge of everything IT. Professor Cook's understanding of the thesis process was the key factor in the completion of this research and subsequent thesis. His ability to motivate and willingness to make time for student thesis work was selfless and far beyond the requirements of his role as a thesis advisor.

The author would also like to thank Dr. Doug MacKinnon for his support as the second reader. His time and insights were invaluable during this process.

Special thanks to Mrs. Susan Hawthorne for the brainstorming sessions that ultimately led to this thesis topic. Her ability to think outside of the box was inspiring.

The author would also like to thank Captain Dorothy Reed, Deputy Chief of Staff(N1) and CDR Rick Tolley, Assistant Deputy(N1B) at CNRFC for the thesis topic suggestions and the support during this lengthy process.

My sincere appreciation and gratitude to LCDR Byron Moss (N63) and LCDR Jeff Bennington (N66) at CNRFC for providing much needed research data and ideas for future initiatives.

A special thanks to LT Fulmer for his expertise and knowledge in creating the WITS Web-Based Application. The Web Application would not have been possible.

The author would also like to give a special thanks to LCDR(ret) Phil Wessel for his mentorship and candid guidance. He would not be here without it.

LT Britt would like to express his love and appreciation to his wife, Mrs. Shannon Britt. Her loving support, limitless patience, and encouragement helped him stay the course during this long and difficult process. The author would also like to express his love to his daughter, Madelynn Britt. Her understanding and "feel better daddy hugs" made it all worthwhile. Finally, LT Britt would like to say thank you to his family and shipmates for all the encouragement and distractions, when he needed them the most.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION AND BACKGROUND

This thesis investigates the advantages and disadvantages of transitioning to Web Applications and Thin Client-Server Architecture for U.S. Navy shore-based Components. Thin Clients and Web Technology have advanced significantly over the last few years and now more than ever, offer a multitude of cost efficient solutions. In the past, networking technology and bandwidth limitations made traditional Personal Computers or “Fat Clients” a more viable option for Naval Commands. The advancements in networking technology and Wi-Fi have significantly reduced these constraints. Moore’s Law has held constant, advancing digital storage and processing capability far beyond the traditional Client-Server Architecture’s ability to take full advantage of these services. The proliferation of server and network technology continues to provide economies of scale that drive down the cost of hardware. The accessibility of these technologies enabled application and software developers to steadily increase the size and complexity of software and applications. The Fat Client’s proliferation led to most of this software and application development in the form of Native Applications. The cost of Software and Native Applications written for Fat Client platforms continues to increase while Server utilization remains negligible. Decentralization due to the inherent local access precipitated by the use of Fat Client-Server architectures and Native Applications creates surplus server capacity and redundant data centers. The Department of The Navy’s (DON) focus is shifting to Thin Clients and Enterprise Software Licensing due to budgetary constraints and the need for increased efficiencies. It is possible that Thin Client-Server Architecture and Web Applications may be able to provide these cost savings and efficiencies.

A. PURPOSE

The DON recognizes the possible benefits of shifting the IT infrastructure of Naval shore installations to centralized services and Thin Clients. The following excerpt was taken from the Navy Administrative Message (NAVADMIN) issued by the Department Of The Navy Deputy Chief Information Officer (DDCIO (N)) on 13 January 2011:

> As an outcome of the 10 Nov 10 Chief Of Naval Operation (CNO) Executive Board And The Department Of The Navy (DON) Information Technology (IT)/cyberspace efficiencies, The Department Of The Navy Deputy Chief Information Officer, Navy, (DDCIO (N)), will work with all stakeholders to implement and explore additional IT efficiencies in the areas of Enterprise Software Licensing (ESL), data centers and thin clients (computing devices that rely on servers to perform data processing). (Dorsett, 2011)

This research was focused on identifying possible IT solutions for shore-based Reserve Component Commands consistent with the goals of the DON and directives issued by the DDCIO (N). Those goals include consolidation of data centers and increased server utilization. The DDCIO (N) has directed the following:

> Each Echelon II Command shall work with the Navy technical authority to develop a plan to reduce data centers by 25 percent, increase server utilization by 40 percent or more (not to exceed 80 percent utilization) and increase server virtualization by 50 percent (not to exceed 80 percent virtualization). This plan shall be submitted to DDCIO (N) by 30 Sep 11. These targets must be realized no later than 30 Sep 12 ..Echelon II'S are encouraged to partner together to maximize rationalization and virtualization of systems and applications, and to regionally consolidate servers and data centers where feasible. Maximum effort should be applied to reduce the it footprint and infrastructure in an effort to save Navy resources in hardware, software, manpower and to promote navy green it efforts. (Dorsett, 2011)

Web Applications, Thin Clients, and Smart Mobile Wireless Devices were identified as possible IT solutions capable of facilitating the DDCIO (N)'s goals. The advantages and disadvantages of transitioning to Thin Client-Server Architectures, Web Applications, and leveraging Smart Mobile Wireless Devices were compared to current solutions. The insights gleaned from this research were then applied to Commander Naval Reserve Forces Command (CRNFC) to determine their potential for helping CRNFC and similar shore commands meet the DDCIO (N)'s stated objectives.

B. RESEARCH QUESTIONS

What are the advantages and disadvantages of using native vs. web-based applications for Navy Commands? This is the principal research question this thesis seeks to answer. IT solutions supporting Thin Client-Server Architecture and the DDCIO (N)'s objectives were also addressed. The potential for increased functionality and efficiencies through the use of mobile technologies was also considered. The following related research questions were essential in providing a complete composite picture for Thin Client-Server Architecture and related technologies as viable cost efficient solutions:

- What are the advantages and disadvantages of Thin Client-Server Architecture for Navy Shore Commands
- How does the Total Cost of Ownership of Thin Clients compare to that of Fat Clients?
- What are the advantages and disadvantages of using mobile native applications vs. mobile web-based applications?
- What type of implementation strategy should Naval Shore Commands employ to prepare for the change?

C. BENEFITS OF THE STUDY

Budgetary constraints are a primary focus of the current administration and Department of Defense as a whole. The U.S. Navy is taking the initiative to proactively seek cost cutting measures and increase efficiencies in every functional area of the enterprise. IT systems, software applications, and their associated infrastructure represent a significant portion of the DON Budget. The Chief of Naval Operations (CNO) and DDCIO (N) identified several key areas and possible solutions to maintaining or increasing IT efficiencies while meeting future budgetary constraints. CNRFC is already taking steps to comply with the DDCIO (N)'s directives. This research will help to further identify future IT solutions to complement their current initiatives. Incorporating support for mobile applications will also be studied as a possible low cost solution to extend mobile device functionality and efficiencies to the Reserve Force.

These findings may then be applied to benefit all Navy shore-based commands in meeting the objectives outlined by the CNO and DDCIO (N).

D. SCOPE

This research will focus primarily on IT systems for shore-based Naval Commands. The research may also apply to IT systems and IT services across the DoD and DON as we transition from current NMCI services to Next Generation (NGEN) IT solutions. The DoD and DON have issued directives concerning the need to increase efficiencies in order to meet budgetary constraints. The DDCIO (N) has tasked IT Managers with actively seeking solutions to consolidate data centers and ensure servers are being fully exploited. Thin Client-Server Architecture and Web Applications have been specifically designated as possible solutions to facilitate DON IT objectives. The advantages and disadvantages of Thin Client-Server Architecture and Web Applications will be researched as they apply at the shore command level. IT Initiatives at CNRFC will be analyzed and incorporated into this thesis research. Native and Web applications developed by the DoD and DON will be compared and included as proof of concept. NPS student developed Web Application will also be analyzed and provided to facilitate further understanding of Web Application development and functionality. The potential IT solutions identified by this research may then be applied by Naval Shore Commands seeking to meet the DDCIO (N)'s objectives.

E. METHODOLOGY

The following methodology will be used to conduct research and analysis. First, data relating to Client-Server architectures, Hardware, Software, and Applications will be gathered during background and literature review. The extensive amount of information available on these technologies should provide the vast majority of data for comparison and analysis. Next, this research will study the advantages vs. disadvantages of Thin Client-Server architecture, Native vs. Web-Based Applications, and Mobile Native vs. Mobile-Web Applications. Total Cost of Ownership will be calculated and compared. Proof of concept Applications will be provided and analyzed. This data will be examined in order to identify the most efficient IT solution to meet the Department of the Navy's IT infrastructure goals. Finally, conclusions and recommendations will be offered based on the research and analysis conducted in Chapters II through IV.

F. ORGANIZATION OF THE THESIS

This thesis will be organized according to the following Chapters:

1. Background Information and Literature Review

Chapter II will acquaint the reader with the history and evolution of applications and their associated hardware. It provides a brief synopsis of the major technological advancements that were responsible for catapulting us into the information age. The next section familiarizes readers with servers and database systems. Computer applications are then defined according to how they interface with the different types of servers and database systems. The subsequent sections define the terms used and introduce the reader to the different types of hardware that will be referenced in later chapters.

2. Survey of Architectures, Technologies, and Applications

Chapter III examines the architecture, technologies, and applications identified in Chapter II. A comparative analysis of the Gartner Research Group's Total Cost of Ownership for Thin Clients and Fat Clients was examined and the findings presented to show the potential for cost savings. The advantages and disadvantages of Thin Client-Server Architecture are analyzed and presented to the reader. Subsequent sections outline the advantages vs. disadvantages of Native vs. Web-Based applications and the associated platforms.

3. Cost Comparison, Current Initiatives and Examples, Student Developed Web Application

The TCO case examined in Chapter III was used as a model to analyze and compare current NMCI Fat Client costs at CNRFC with industry standard Thin Clients at CNRFC. The cost of Web Access at CNRFC was then used in conjunction with Thin Client costs to extrapolate the cost of moving to Thin Client-Server services at Naval

Shore Commands. Native Applications developed by the Office of Naval Research and other DoD organizations were downloaded and analyzed for comparison to current Web Applications available through the DoD/DON. Finally, a Web Application developed by LT Britt and LT Fulmer was presented to show the potential for Web Applications and future development possibilities.

4. Conclusions and Recommendations

Chapter V presents a summary of previous chapters and analysis. Findings are summarized and condensed for the reader. The benefits for the DON and Naval Shore Installations will be discussed along with suggestions for current application to meet DDCIO (N) objectives. Finally, future areas for study will be recommended.

II. BACKGROUND INFORMATION AND LITERATURE REVIEW

A. BACKGROUND

The advent and discovery of advanced numbering systems led to the need for advances in technology to support their use in a variety of applications. The Babylonians used a positional number system that employed base 60 in order to create cuneiform tables they could use to perform multiplication, division, and solve algebraic equations (O'Regan, 2008). These tables were then used in astronomical and advanced problem solving applications. Among the most influential contributors to modern western civilization were the Greeks. Not only were they the world's first democracy, but their introduction of mathematics in 500 B.C. led to the discovery of the Euclidean algorithm (O'Regan, 2008). Algorithms are the fundamental building blocks of modern day computer applications.

The introduction of Boolean algebra by George Boole in his paper (Mathematical Analysis of Thought) and book (An Investigation of the Laws of Thought) in the 19th century, became the cornerstone of modern binary digital computers (O'Regan, 2008). Charles Babbage is also considered a founding father of modern computer science. He used his knowledge of mathematics and mechanics to design the Difference Engine and later the Analytic Engine (O'Regan, 2008). Babbage adapted the latter design to use punch cards, which he modeled after the Jacquard loom. These cards could automate control of the machine and instruct the analytic engine to perform a multitude of calculations using operation and variable cards. His design is thought to be the first mechanical computer and represented a huge leap forward in computer science (O'Regan, 2008). This led to Claude Shannon's work with Boolean Algebra and its application in circuit design. He applied this knowledge in his Master's thesis, "A Symbolic Analysis of Relay and Switching Circuits," which is the quintessential groundwork for all modern computer and telephone switching circuits in use today (O'Regan, 2008). The next leap forward came with Jon von Neumann's concept of a single data store incorporating both programs and instructions, which he called the Neumann architecture (O'Regan, 2008). This allowed for the use of punched cards to perform different operations. Most computers in the twentieth century were based on the von Neumann architecture (Goldstine & Goldstine, 1996).

B. EVOLUTION OF COMPUTERS

The ENIAC was the first device developed using Jon von Neumann's architecture by a team of personnel, most notably Eckert and Mauchly. Its development was commissioned by the Army Ordnance Department in an effort to accelerate and automate calculating and producing firing tables for use during WWII (Goldstine & Goldstine, 1996). It required the use of punched cards and vacuum tubes to perform the intended operations, while general applications required the circuitry to be re-wired (Goldstine & Goldstine, 1996). The next evolution was the EDVAC, which took advantage of the new von Neumann architecture. The EDVAC was designed with the ability to store data and command instructions in the same memory location allowing for a machine that was more easily re-configured. In 1948, the Manchester Mark I computer was designed in England utilizing the new von Neumann architecture to its full potential. The first prototype executed operations via a program stored in memory in 1948 (O'Regan, 2008).

During their work on the ENIAC, J. Presper Eckert and John Mauchly envisioned a computing machine designed for use by mathematicians, engineers, scientists, and businesses alike. They developed the UNIVAC and in 1951 delivered the first functioning machine to the U.S. Census Bureau (Ceruzzi, 2003). Their incorporation of a magnetic tape media storage system, stored program architecture, and fewer vacuum tubes made the UNIVAC faster and more reliable. The U.S. Air Force was the first "customer" to have a fully functioning UNIVAC delivered and permanently installed in 1952 (Ceruzzi, 2003). By 1954, Eckert and Mauchly managed to sell and deliver twenty of the machines to various government and private organizations (Ceruzzi, 2003). The machines were used to process vast amounts of data in the form of engineering calculations, U.S. population tables, governmental logistics problems, and inventory control. Thus began the data processing revolution and the evolution of the machines that would automate these processes and applications.

C. MAIN FRAMES TO PERSONAL COMPUTERS

The ENIAC used a system of punched cards and relays to store and transfer program instructions. The cards were fed into an IBM card reader and then transferred to the relay system (memory) via a constant transmitter (Goldstine & Goldstine, 1996). The system was extremely reliable when set up correctly, but also extremely slow and difficult to re-configure for different applications. The UNIVAC built on these fundamentals, but Eckert and Mauchly's utilization of the stored-program and magnetic tape helped usher in the era of commercial computing (Ceruzzi, 2003). As the popularity of these systems grew, so did the amount of data and desire to run increasingly complex applications. The main limitations of these early machines were the high cost, size of the systems, and the lack of large capacity data storage. Magnetic Drum technology offered substantial increases in data storage, but at the detriment of access time. The development of core memory was the first step toward the realization of cost-effective commercial applications:

> Core memory refers to small, doughnut-shaped pieces of material through which several fine wires are threaded to store information. The wires passing through the core can magnetize it in either direction; this direction, which another wire passing through can sense, is defined as a binary zero or one (Ceruzzi, 2003, p. 49).

The advantages of this configuration over tubes and magnetic strips are: miniaturization, nonvolatile memory, and RAM ability. RAM allows for the access of any bit stored in the core to be accessed and retrieved as quickly as any other bit (Ceruzzi, 2003).

The advent of transistors was the final innovation needed and led to huge leaps in computer circuitry. They replaced vacuum tubes and allowed for significant reductions in size and complexity. Their proliferation as the fundamental components of processors drove down production costs and allowed for development of processing circuits that could take full advantage of advances in memory. These transformations were used by

IBM to build the archetypal main frame computer, the model 7094 (Ceruzzi, 2003). The term “main frame” refers to the physical configuration of circuits, which are mounted on metal framework and stored in large cabinets. Most main frames were designed for commercial use as evidenced by the cost of IBM’s model 7094. In 1963 it cost $30,000 a month to rent or $1.6 million to purchase the model 7094 (Ceruzzi, 2003). The main frame’s cost precluded it from use in all but the largest of corporations and government entities. In conjunction with the prohibitive cost was the main frames configuration. The 7094 was designed to process batch files and lacked a video console for direct access to data (Ceruzzi, 2003). This was typical of most main frames due to the extreme expense associated with allowing the processor to sit idle while individual programs were loaded and executed. Despite the excessive costs, the benefit of automating business applications became very apparent and soon businesses and government at all levels were eager to use this technology (Ceruzzi, 2003).

By the latter part of the 1950s, transistor technology matured to a point that allowed IBM and others to offer solid state machines to the emerging small business market. These machines were smaller, cheaper, and almost as capable due to advances in solid state technology. These machines were no longer main frames, but fully capable miniaturized versions targeting business customers. One of the most successful machines built and sold was the IBM 1401. By the end of the 1950s, approximately forty UNIVACs had been sold compared to over ten thousand 1401s (Ceruzzi, 2003). This emerging market grew even faster with the dawn of the space race in 1961. The advent of assembly languages like FORTRAN and COBOL made utilizing these machines easier and more efficient. COBOL became one of the first standardized languages due to the insistence of the U.S. DoD that it be used in all computing equipment purchased or leased by the U.S. government. In 1965, the Digital Equipment Corporation shipped the first PDP-8 and launched the era of the “Minicomputer” eventually selling over 50,000 units (Ceruzzi, 2003). The minicomputer firmly established the market for cheaper miniaturized computing technology.

The evolution of the "Personal" Computer began with the Digital Equipment Corporation's PDP-10. The PDP-10 was still considered a mainframe in size, but had a random-access disk and allowed the use of DEC tape. The random-access disk allowed users direct access to their files and the DEC tape allowed users to load "personal" files and programs (Ceruzzi, 2003). In 1972, MIT helped Digital Equipment Corporation develop the TOPS-10 system. TOPS-10 allowed users to manipulate files from a terminal during their allotted user time. It also allowed them to output the contents of a file to an output device of their choice, such as a printer or teletype (Ceruzzi, 2003). This type of access and control gave users the feeling that the PDP-10 was their own personal machine during that time frame. These machines were sold to corporations that leased computer time, one of which was Computer Center Corporation or C-Cubed. C-cubed gave, a then teenage Bill Gates, free time on the computer in exchange for finding and fixing system bugs (Ceruzzi, 2003). These experiences showed users the potential of "personal" computing.

The next giant leap in computing technology was forecast by Gordon Moore in 1964. He observed that the number of transistors placed on a single integrated circuit had doubled every year since its invention (Ceruzzi, 2003). This doubling effect led to the creation of the integrated circuit. Moore's Law and increasing chip density led to some of the first truly "personal" devices in the form of calculators. One of the first calculators to take advantage of electronics was developed by Wang Laboratories, the Wang LOCI. Hewlett-Packard developed the HP-9100A a few years later and sold it for around $5,000 (Ceruzzi, 2003). These calculators were smaller than their mechanical counterparts, but still about the size of a personal computer today. In 1971, Bowmar advertised the Bowmar Brain just in time for Christmas. It used integrated circuits, was the size of a paperback book, and cost around $250 (Ceruzzi, 2003). In 1972, Hewlett-Packard took it a step further and developed the first pocket calculator selling for $400 (Ceruzzi, 2003). Hewlett-Packard introduced the first programmable pocket calculator in 1974, the HP-65. These calculators could store keystroke sequences in memory which could be used to write limited programs. Hewlett-Packard advertised these as a "personal computer" (Ceruzzi, 2003).

The introduction of the programmable calculator led to the first consumer market for logic chips and the economies of scale that followed. These electronic devices enabled individual users to explore their own "personal" avenues of creativity. The pocket calculator showed the potential for integrated circuits, but their chips were still too specialized for a general-purpose computer (Ceruzzi, 2003).

Intel produced a set of four chips in 1971 called the 4004 that included, "a micro programmable computer on a chip!" (Ceruzzi, 2003, p. 220). This set of chips contained a general-purpose stored-program computer, Random Access Memory (RAM) chip, Read Only Memory (ROM) chip, and an Erasable Programmable Read-Only Memory (EPROM) chip. It was capable of processing 4 bits at a time. The ROM chip supplied the code necessary to make the general-purpose processor specific to a customer's needs (Ceruzzi, 2003). The EPROM chip could be erased and reprogrammed using ultraviolet light making it a fundamental part of system design using a microprocessor. A year or so later, Intel introduced the 8080 which could process 8-bits at a time or a full byte. It was the first microprocessor that approached the minicomputers processing capabilities (Ceruzzi, 2003).

In January of 1975, H. Edward Roberts designed the Altair (Ceruzzi, 2003). It was an inexpensive computer designed around Intel's 8080 microprocessor. Popular Electronics ran an article advertising the Altair for less than $400. It was not long before hobbyists were clamoring for the device kits. MIT began building completed Altairs for $498, but soon became backlogged with orders. Companies offering plug-in boards for added functionality and memory began to spring up as its popularity increased. One of the major limitations was that the Altair lost its data when power was removed (Ceruzzi, 2003). This was remedied when David L. Noble at IBM developed the floppy disk for the System/370, which was later adapted for the IMSAI 8080 version of the Altair. This version was developed by Digital Research which was founded by Gary Kildall and Dorothy McEwen. It was the first truly Personal Computer with a video monitor, disk storage, Bill Gates BASIC operating system, and Kildall's Basic Input/Output system (BIOS) (Ceruzzi, 2003). Figure 1 depicts all the components of the IMSAI 8080 version of the Altair.

Figure 1. IMSAI 8080 version of the Altair (From Ceruzzi, 2003)

From the creation of the ENIAC to the proliferation of the Personal Computer, U.S. DoD projects and commercial funding continue to drive the evolution of computing technology. We now have computing devices with capabilities that far exceed the original main frame and yet small enough to fit in the palm of your hand. The next section defines and describes the major hardware components that provide backend functionality.

1. Web Server

The Web server functions as a service delegation center (Sintes, JavaWorld.com, 2002). Web servers accept incoming client requests and select the most appropriate server-side application to pass it to for processing. The server-side program processes and executes these requests, which are then passed back to the Web server and sent to the client. Web servers are designed to work primarily with Hypertext Transfer Protocol (HTTP) (Sintes, JavaWorld.com, 2002). In the basic configuration, a client sends an HTTP request over the network to the web server. The web server processes the request and sends back a static Hypertext Markup Language (HTML) page, redirects the client to another server, or initiates a request to another server-side technology. Requests to other server-side technologies are commonly processed by programs such as Common Gateway Interface scripts, Java Server Pages, servlets, Active Server Pages, server-side JavaScripts, or other server-side programs that provide a response using HTML for display on the client's web browser (Sintes, JavaWorld.com, 2002).

2. Application Server

The basic Application server gives client applications access to business logic using a variety of protocols. Application servers communicate with clients using program logic in the form of data and method calls. The client applications then use the business logic to produce the appropriate output. The server uses component Application Programming Ineterfaces(APIs) like Enterprise JavaBeans Technology(EJB) to expose the business logic to the client (Sintes, JavaWorld.com, 2002). Since Application servers return the data requested as application logic, the data can be re-used by multiple client types employing different applications. Resource management such as security, transaction processing, resource pooling, and messaging are administered by the Application server itself (Sintes, JavaWorld.com, 2002).

3. Database System

Database is defined as, "an ordered collection of related data elements intended to meet the information needs of an organization and designed to be shared by multiple users" (Ponniah, 2005, p. 19). In a database system an ordered collection refers to the way in which data is configured and linked to create a logical data structure. Relations among these data elements are created that represent the organizations information needs. Depending on the number of users that need access, a database can be configured to support multiple authorized users.

Organizations must decide whether to employ a Process-Driven or Data-Driven approach (Ponniah, 2005). Both methods require the organization to first define requirements. After the definition of requirements has been met, the two methods deviate. The process driven method uses outputs and processes to determine inputs.

Once inputs have been identified, data files are designed according to the final set of requirements. The data-driven method requires the collection of data about business object requirements. Next, the database structure is designed. The last step is the design of initial processes. The data driven approach allows organizations the flexibility to add further processing requirements in the future (Ponniah, 2005). This is the approach most often employed by organizations today (Ponniah, 2005).

A typical database system consists of the following basic elements: data repository, data dictionary, database software, data abstraction, data access, and transaction support (Ponniah, 2005). The data repository consists of the physical storage where all of the data files are kept. The data dictionary is made up of the structures and relationships among the different data elements. The data dictionary and data repository interact to provide relevant data to end users. Database software or database management software (DBMS) supports the administration of the database. It allows ease of management through features like storing, retrieving, and updating the data in a database system. Data abstraction is, "the ability to hide the complexities of data design at the levels where they are not required"(Ponniah, 2005, p. 22). This allows the database administrator to hide field sets and data structures that a customer does not need or does not care to see (Ponniah, 2005). It means the database can be configured to output only the level required by the different user groups. Basic data access is achieved using four fundamental operational commands listed as follows: READ, ADD, UPDATE, and DELETE (Ponniah, 2005). Transaction support refers to a database systems ability to complete operational commands/functions while still maintaining the integrity of the database. This is accomplished by allowing a transaction to complete entirely or terminate and rescind data updates due to malfunctions that prevent completion (Ponniah, 2005).

Database systems are organized according to the data requirements of the organization. Data models are used to represent the real-world data requirements of the organization. Some of the most common data models are Hierarchical, Network, Relational, and Object Oriented databases (Ponniah, 2005).

- Database systems use parent-child relationships between pairs of data structures in order to provide a logical, searchable structure (Ponniah, 2005). In this type of structure a child can only have one parent, but a parent can have multiple children. This type of relationship is also known as one-to-one in the case of child-parent relationships or one-to-many in the case of parent-children relationships. The structure relationships determine how queries are written to search the database (Ponniah, 2005).
- The many-to-many relationship allows data structures to be linked to each other without restrictions. These relationships are also referred to as owner-member relationships in which an owner can have many members and a member can have many owners (Ponniah, 2005).
- Hierarchical data models have levels and each level contains data structures representing business objects. The data structures are assembled in parent-child relationships that exist as one-to-one or one-to-many. Hierarchical models have a root segment which is typically the data segment at the top level of the hierarchy (Ponniah, 2005). The levels and parent-child relationships are linked via physical pointers or physical storage addresses that are embedded in the records of the database (Ponniah, 2005).

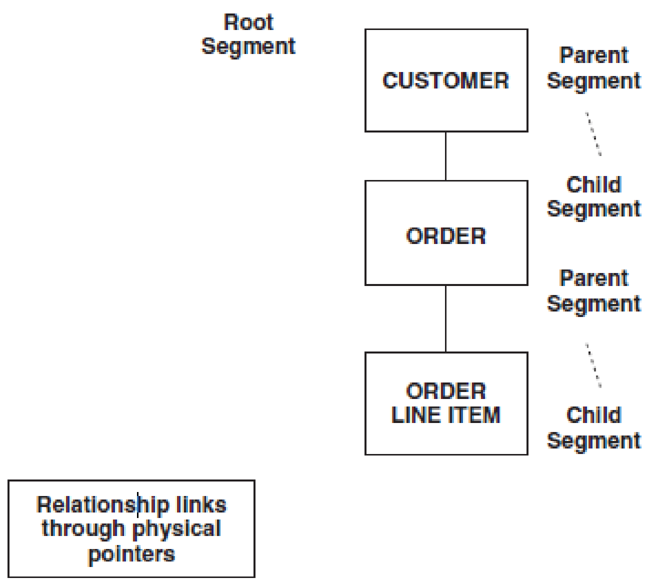


Figure 2. Hierarchical Data Model (From Ponniah, 2005)

- Network data models do not conform to the hierarchical data model, they are more indicative of real world data structures (Ponniah, 2005). Network data models allow many-to-many relationships at any level. The levels in the network model are structured without restriction. Data structures within successive levels are connected to any other dependent structures where appropriate. Every data structure in the network model is known as a record type (Ponniah, 2005). Relationships are established among record types by designating one the owner record type, with member record types linked accordingly. A member record type may have many owner record types and owner record types may have many member record types. Each member and related owner type form a set (Ponniah, 2005). Physical pointers connect related occurrences of record types using pointers or physical storage addresses embedded in the database records.

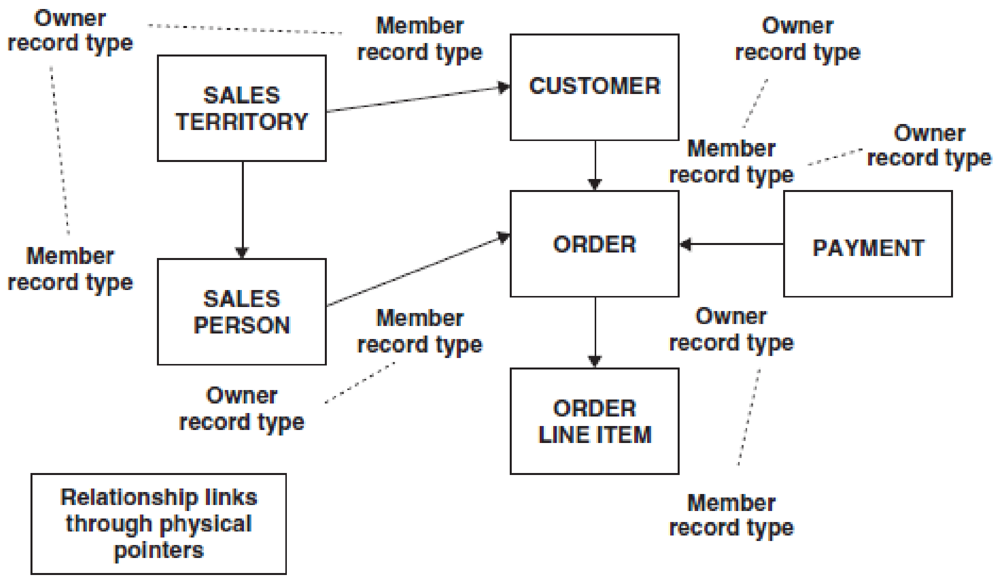


Figure 3. Network Data Model (From Ponniah, 2005)

- Relational data models employ a similar methodology as network models for level structure and their associated relationships (Ponniah, 2005). However, network models and hierarchical models require the use of pointers to make related data connections. Ponniah stated, "This is a serious drawback because you have to rewrite the physical address in the data records every time you reorganize the data, move the data to a different storage area, or change over to another storage medium." (Ponniah, 2005, p. 27) The relational model uses foreign keys as logical links to make the connections to all related instances or occurrences. The foreign key field is then used to search or query the database looking for any instances that contain that particular key. In this way the relational model supports both one-to-one and many-to-many relationships.

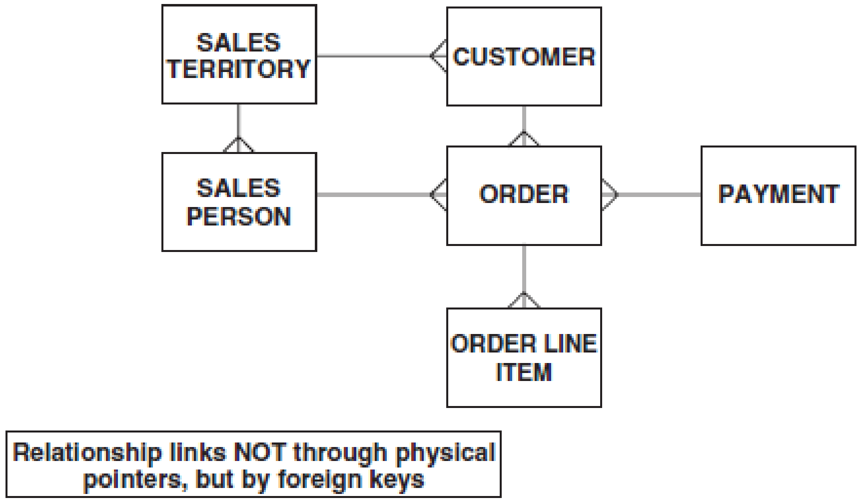


Figure 4. Relational Data Model (From Ponniah, 2005)

- Object-oriented databases take complex systems and allow applications to be constructed with reusable components using advanced software modeling and development methods (Ponniah, 2005). Object-oriented databases support Binary Large Objects (BLOBS) and allow users to define their own data types. These models combine data and process definitions and enable users to define their own functions. Object-oriented models improve the developer's ability to represent real world entities (Ponniah, 2005). There are three basic principles that enable the object oriented model to accomplish complex data manipulation and data modeling: data abstraction, object identity, and inheritance (Ponniah, 2005). Data abstraction refers to identifying the important aspects of what an object is and what it does and separating those from those considered lower priority. This is done by encapsulating the important aspects within the definition of the object. In this way, the object houses both the data structure and the operation set. Abstraction also allows for information hiding by using abstract data types (ADTs) for the encapsulation. Only the user interface and processing requirements are visible to users and programs, the internal details are shielded from the rest of the system. Inheritance allows objects in the model to inherit code and structure from other higher level entities within the database, greatly simplifying database execution. Finally, object identity allows one object to differentiate another using its attributes. In this way objects may refer to or contain other objects.

D. COMPUTER APPLICATIONS

1. Native Applications

Native applications are applications written and compiled using code native to the device (Charland & LeRoux, 2011). These applications require client side processing capability. The applications are written using platform specific code in order to optimize the performance of the application on that particular device. This allows programmers to ensure that data is displayed based on precise parameters like "physical size, color depth, screen resolution, pixel density, and aspect ratio." (Charland & LeRoux, 2011) Other platform considerations are user input type and device capabilities (Charland & LeRoux, 2011). Native Applications can be tailored by IT departments according to a user groups specific needs. The applications are setup to use the client server architecture employed by that organization or service provider.

2. Web Applications

Web Applications are written in interpreted languages like Javascript and then accessed using the devices built in web browser (Charland & LeRoux, 2011). The data is displayed on the device through the use of HTML and Cascading Style Sheets (CSS). This type of application relies on the Web Server and Application Server for processing. Any device with a web browser can access and use this type of application as long as they have access to the network where the servers are housed. This type of application is not dependent on the type of architecture used by the enterprise.

E. CLIENT SERVER ARCHITECTURE

Client server architecture has many forms. The basic architecture consists of a personal computer or client that requests information or data from a server via a network (Urbanowicz, 2001). The Encyclopedia of Computer Science defines client server computing as:

> A distributed computing model in which *client* applications request services from *server* processes. Clients and servers typically run on different computers interconnected by a computer network. Any use of the Internet such as information retrieval from the World Wide Web is an example of client-server computing. However, the term is generally applied to systems in which an organization runs programs with multiple components distributed among computers in a network. The concept is frequently associated with *enterprise computing*, which makes the computing resources of an organization available to every part of its operation. ("Client Server," 2003)

Organizational needs as well as existing infrastructure, budget, and IT staff expertise have huge implications on how this architecture is employed. There are four main ways in which client server architecture is configured:

- "Graphical User Interface (GUI) front-end on legacy systems, with the client having a graphical user interface to an existing legacy application" (Daniel, 1996).
- "Distributed application, with the application on both the client and server, and data on the server" (Daniel, 1996).
- "Remote data management, with the application on the client and data on the server" (Daniel, 1996).
- "Distributed database, with the application and some data on the client, and data on the server" (Daniel, 1996).

These configurations are most often implemented using the following: host-client environment, basic client-server environment, client-server 2-tier environment, client-server multitier environment, client-server multitier distributed, client-server/Web server, and Thin Client-Server (Urbanowicz, 2001).

1. Host-Client

The host client environment consists of a mainframe and thin client often referred to as a "dumb" terminal. The pros of this type of system are the ease of client installation and the reliability of proven main frame technology (Urbanowicz, 2001). The cons are the high cost of the system, the high cost of maintenance, and the inflexibility of using older technology (Urbanowicz, 2001).

2. Basic Client-Server

In the basic client-server environment, the client queries the server and waits for the server application to process the queries, analyze, and retrieve the data. The data is sent back to the client via the network and displayed on a GUI via the client application or on a host terminal. The pros are the ability to host multiple users and display data via a GUI. The cons are software deployment, software management, and over utilization of the client (Urbanowicz, 2001).

3. Client-Server 2-Tier

The client-server 2-tier architecture involves incorporation of a DBMS to provide additional processing in a traditional client-server architecture. This allows some of the client side processing to be reduced and thus the need for extensive client side software deployments (Urbanowicz, 2001). Since the server contains most of the software, changes are made mainly to the server itself reducing maintenance and labor costs. The pros are the reduced client side software deployment costs and the increase in performance due to less processing by the client. The con is the need for increased processing power on the server side and the costs associated with it (Urbanowicz, 2001).

4. Client-Server Multitier

Client-server-multitier architecture employs a middle tier of centralized processing servers. Multiple database servers store the data for the multitude of applications required by large organizations. The middle tier is used as central support for each type of database server by incorporating shared business rules and software models. The centralized location of the middle tier facilitates change management and standardization that can be easily migrated. The pros of this configuration are the ability to balance performance across the network, shared reusable rules, and standardization (Urbanowicz, 2001). The cons are the increased software costs of using multiple applications, the cost to maintain several vendor relationships, and complex infrastructures (Urbanowicz, 2001).

5. Client-Server Multitier Distributed

Client-server multitier distributed allows for multiple configurations throughout an enterprise. This type of architecture provides a further level of tailoring and performance balancing. This is accomplished using centralized processing servers in conjunction with local servers that process the data sent from the centralized server hub and forward to clients accordingly. The pros are load balancing, data transmission control, increased network efficiency, and the ability to tailor services within a multifaceted enterprise (Urbanowicz, 2001). The cons are the increased expense due to additional servers, increased manpower costs, data security, and extremely complex infrastructure (Urbanowicz, 2001).

6. Client-Server/Web Server

Client-server/Web server can be set up in the form of a second tier, middle tier, or n-tier architecture (n represents the number of tiers i.e., 3-tier, 4-tier, etc.) (Urbanowicz, 2001). A web server accepts requests from the client and acting as a form of client itself, forwards them to a database server that fills the request. The web server then acts as the second tier and processes the data which it then forwards to the client for display. The web server can also be set up to send embedded applications that run on the client's web browser. This is primarily used for managing web content such as video, pictures, and other media formats (Urbanowicz, 2001). The pros are easy client deployment and online access from any location. The cons are the increased costs of equipment and labor.

7. Thin-Client Server

Thin-Client Server architecture may be employed using multiple forms of the traditional Client-Server architecture. The biggest difference is the addition of thin clients, which shift the application processing to the server. The Servers allocate hardware and software resources to each Thin-Client depending on the type of applications needed by the user (Machovec, 1997). The pros are: efficient allocation of resources, centralized software management, and decreased cost per client device. The cons are: increased costs of servers, reliance on network access, and user perception.

F. FAT CLIENT

Fat clients are typically personal computers used in a client-server architecture where application processing is done on the PC and data retrieval on the server (David, 2003). These systems are typically made up of a monitor, tower, keyboard, mouse, and other peripheral devices. The tower holds the motherboard and main processor as well as video and audio cards. This enables the fat client to handle robust applications requiring significant application processing. These systems are capable of allowing a user to access applications and process data even when network connection is unavailable. In this configuration, they can be considered standalone workstations until the network is available or access is required.

G. THIN CLIENT

A thin client is a solid state device that provides a connection to the thin-client computing environment via the network. The thin-client computing environment is hosted on a centralized server that does the application processing and data retrieval

(Becta, 2004). This is the basic premise of what constitutes a thin client. Using this basic premise, several thin client types emerge based on the type of client network employed by the organization.

1. Ultra Thin Client

These systems consist of an extremely thin form factor client requiring a keyboard, mouse, and a monitor. These clients are typically used in the host-client environment where all processing with the exception of keyboard processes and screen output are completed by the server (Becta, 2004). The lack of hard drive, expansion disks, disk drives, and memory cards contributes to their extremely light weight, lower costs, and minimal power consumption. Figure 5 depicts a typical Ultra Thin Client.

Figure 5. Sun Ray 3i Client (From "Sun Ray virtual display clients," 2011)

2. Windows-Based Terminals

There are two types of terminals designed to work with the Windows Operating System and associated software. The first uses the Windows-Based Terminal Standard in conjunction with Microsoft Remote Desktop Protocol (RDP), Citrix Independent Computing Architecture (ICA), or VMware to display the windows environment on the desktop (Becta, 2004). Figure 6 is a Wyse WBT capable of operating with a traditional PC or as a Thin Client in conjunction with a monitor.

Figure 6. Wyse C90LEW (From "Wyse Technology - Wyse C90LEW thin client," 2011)

The second type consists of proprietary client operating environments supported by Citrix ICA and RDP to display the windows environment on the desktop (Becta, 2004).

3. Low Spec PC Solution

This thin client solution uses an organizations existing "fat clients" in conjunction with a thin client network to provide enhanced network capability without excess hardware expenditures (Becta, 2004). The existing PCs are linked to the thin client network using web-based terminals or a basic thin client software solution allowing them to tap into new software and applications via the thin client network. This solution offers cost savings, extends the life of existing legacy equipment, and provides a transition alternative.

4. Tubby Clients

This is similar to the low spec PC solution. However, tubby clients refer to newer PCs still well within their expected useful life. They have functioning operating systems and may be configured to run applications in a traditional tiered client-server environment or via thin client software. This allows the organization to run legacy applications that may be incompatible with thin client networking, while still allowing users access to newer applications via the thin client network (Becta, 2004).

H. SMART MOBILE WIRELESS DEVICES

First, what constitutes a "smart mobile wireless device?" Merriam Webster defines smart as, "operating by automation" ("smart," 2011). In this case, the term smart references a devices ability to automate the use of applications via a Graphical User Interface (GUI). Webster defines mobile as, "capable of moving or being moved." ("mobile," 2011) Mobile devices are anything that has a small enough form factor to be easily transported and can operate via an internal power source while on the move. Webster defines wireless as, "having no wire or wires; specifically: operating by means of transmitted electromagnetic waves." ("wireless," 2011) Therefore, for the purposes of this thesis, smart mobile wireless devices are any device with a transportable form factor that can be used to access applications on the go and have the capability to transmit and receive data wirelessly.

Smart Mobile Wireless Devices use wireless cellular networks and Wireless Fidelity (Wi-Fi) to transmit data over a network. Wireless cellular networks either use Packet-switched networks or Circuit Switched networks (Dean, 2006). The various cellular providers use the "xG" nomenclature when referring to their particular commercial cellular network. The x represents the numerical designation and the G stands for Generation, so the first generation of mobile telephony is referred to as (1G). The International Telecommunications Union (ITU) established the standard and nomenclature in the 1980s (Phifer, 2000). The ITU maintains and upholds the standard and is the final authority of what officially constitutes each successive generation in regards to technology and data speeds (Phifer, 2000).

The first deployed commercial cellular networks known as "1G" were Advanced Mobile Phone Service (AMPS) cellular networks (Phifer, 2000). These networks used Frequency Division Multiplexing Access (FDMA) to transmit and receive analog voice over the 800 MHz frequency band. The second generation or "2G" networks split mobile carriers into to two groups. A large number of operators in the United States selected the IS-95 standard or Code Division Multiple Access (CDMA) using the 800 MHz frequency band (Phifer, 2000). Others in the United States, and the rest of the world, chose the

Global System for Mobile communication standard (GSM) using Time Division Multiple Access (TDMA) in the 900 and 1800 MHz frequency bands (Phifer, 2000). CDMA allowed carriers to multiplex or combine 64 calls into a single channel, while TDMA combined 8 calls per channel. The third generation or "3G" standard was formed in 1998 and split into two separate subcategories: 3rd Generation Partnership Project (3GPP) and 3rd Generation Partnership Project2 (3GPP2). The GSM carriers fall under the standards for 3GPP, while CDMA carriers fall under 3GPP2.

There are two main types of network switching used by mobile wireless devices, Circuit Switching and Packet Switching (Dean, 2006). Circuit Switched networks establish a continuous connection between two nodes before data is transmitted. The connection is allocated bandwidth until the connection is terminated. All of the data transmitted uses the same path or circuit selected by the switch until the connection is severed. This uses a significant amount of bandwidth and is slower than packet switching, but provides a stable connection for less tolerant applications like live audio or videoconferencing (Dean, 2006). Packet switching breaks the data into sections or packets before transmitting. Each packet contains information detailing the total number of packets, where each falls in the sequence, and the final destination address. This allows each packet to be transmitted separately using the most efficient path over the network. Connections are not permanently established decreasing bandwidth requirements. Related packets may travel several different circuit paths decreasing network latency. Packet Switching is optimal for transmitting typical network data such as email, application data, and software programs (Dean, 2006).

Many Smart Mobile Wireless Devices are Wi-Fi or 802.11x capable. The Institute of Electrical and Electronics Engineers (IEEE) created this standard in 1997 ("Wi-Fi Alliance: Discover and Learn," n.d.). They established the 802.11 committee to oversee the standard and determine how future specifications would be managed. The 802.11 standard denotes which technology version is in use with a subscript lowercase letter. To date there are four existing Wi-Fi standards: 802.11a, 802.11b, 802.11g, and 802.11n. All four use half-duplex signaling in which a wireless station can transmit or receive, but cannot do both at the same time. Each version is has its own distinct performance, frequency, and bandwidth specifications. The following figure lists them in order of generation (“Wi-Fi Alliance: Discover and Learn,” n.d.).

Wi-Fi Generations

Wi-Fi Technology	Frequency Band	Bandwidth or maximum data rate
802.11a	5 GHz	54 Mbps
802.11b	2.4 GHz	11 Mbps
802.11g	2.4 GHz	54 Mbps
802.11n	2.4 GHz, 5 GHz, 2.4 or 5 GHz (selectable), or 2.4 and 5 GHz (concurrent)	450 Mbps

Figure 7. Wi-Fi Specifications (From “Wi-Fi Alliance: Discover and Learn,” n.d.)

1. Two-Way Pagers

These mobile devices are primarily designed for sending and receiving text messages (Mallick, 2003). They have no voice communication ability and most are not capable of utilizing enterprise type applications. Some of these devices do have HTML browsers and are designed to run data capturing applications. These devices tend to use packet switched networks that do not require constant network connection. It also means that they can access an application without being explicitly connected to the network and transmit data only when required. Due to this and the limited capabilities, they have exceptional battery life (Mallick, 2003). These devices have declined in popularity due to the advent of multi-function devices and their limited capabilities. Figure 8 is an example of a two-way pager made by Research In Motion (RIM).

Figure 8. RIM 857 (From "Research In Motion," 2009)

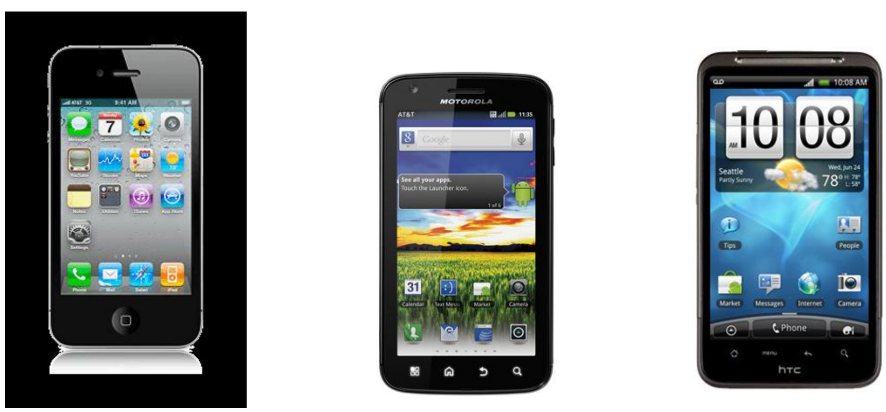
2. Web-Enabled Phones

Web-enabled phones are cell phones that have limited data capabilities (Mallick, 2003). These devices function primarily as voice communication devices capable of running basic targeted web applications. These applications are generally limited to retrieval and display of consumer market data such as weather, stocks, news headlines, etc. (Mallick, 2003). Their limited display sizes and small storage capacity preclude them from most enterprise type applications. They are capable of simultaneous data and voice communications via Wireless Telephony Application (WTA) and they are well suited for text messaging. Another benefit of their limited data capabilities is a longer battery life (Mallick, 2003). Figure 9 is a Web-Enabled Phone made by Samsung.

Figure 9. Samsung Convoy SCH-U640 (From "Samsung Convoy | Samsung SCH-u640 - Cell Phones," n.d.)

3. Smart Phones

These devices represent the new standard of smart mobile wireless devices. They combine the performance and capability of a fat client with the mobility and smaller form factor of a cellular device. These devices have significant storage capacities, RAM, processors, their own operating systems, and exceptional battery life (Mallick, 2003). They are capable of running both enterprise level native and web-based applications. Most are capable of simultaneous data and voice transmission. Most are touch screen enabled devices. The dual nature of these devices allows for the use of applications without network access. There are a significant number of devices available due to their increasing popularity in the commercial sector. Figure 10 shows three examples of 3G capable Smart Phones. The three from left to right are: Apple Iphone4, Motorola Atrix, and the HTC Inspire.



Apple Iphone4 Motorola Atrix HTC Inspire

Figure 10. Smart Phone Examples (From PDAs & Smartphones - Wireless From AT&T, 2011)

4. Tablet PCs

Tablet PCs combine the features of a laptop PC with the addition of a touch screen. They represent the transition from conventional laptops to devices conceptually similar, but with additional features designed to enhance the user experience (Mallick, 2003). They use standard operating systems written for laptops with additional software installed to allow the OS to interface with the touch screen. Specific applications native to the OS are installed that allow the user to take full advantage of touch screen features, like journal software that converts handwriting to standard text (Mallick, 2003).

5. Smart Pads

Smart pads are not a new concept. These devices are very similar to their tablet PC predecessors. They combine the features and processing power of a laptop with increased functionality due to the addition of a touch screen. The biggest difference is the elimination of a physical keyboard in lieu of a software generated keyboard utilizing the touch screen. Most of these devices are WI-FI and 3G capable, eliminating the need for access to data via external media readers and storage devices. This has allowed the elimination of mechanical disk drives, optical drives, and on some models peripheral device ports. Smart pads have significantly smaller form factors than tablet PCs due to the elimination of mechanical drives and use of solid state memory. Figure 11 shows two popular Smart Pad models: Apple Ipad 2 (Wi-Fi capable) and the Samsung Galaxy Tab (3G capable).

Apple Ipad 2 with WI-FI　　　　Samsung Galaxy Tab 3G

Figure 11. Smart Pad Examples (From Best Buy - Computers, Video Games, TVs, Cameras, Appliances, Phones, 2011)

6. Notebook/Laptops

These devices are capable of supporting the same client server applications used on PCs as well as mobile applications used on hand held devices (Mallick, 2003). Laptop and notebook computers are closest in form factor to Tablet PCs. They generally do not have touch screens and are thinner and lighter than their Tablet PC cousin. These devices support the same operating systems as desktop PCs and are able to use virtually any type of application developed by IT departments. Their larger size and shorter battery life make them less mobile and convenient than hand held devices, but their processing power and versatility can make them a preferred enterprise solution (Mallick, 2003).

I. SUMMARY

The discovery and use of advanced number systems in problem solving lead our ancestors to create applications and hardware to aid them in advanced tasks. George Boole and the introduction of Boolean Algebra revolutionized applications and became a cornerstone of modern binary digital computers. The principles of modern computer science were laid by Boole, Babbage, and Von Neumann. These principles were used by Eckert and Mauchly to create the ENIAC, the first mainframe computer. The benefits of the ENIAC facilitated evolution of hardware and software by a multitude of developers. The invention of the transistor furthered this development and contributed to the creation of the integrated circuit. Intel's research and development of the integrated circuit lead to the creation of the microprocessor, which was ultimately used to create the first "personal" computer. This chapter then identifies and defines the applications, hardware, and architectures that have evolved since the introduction of computing.

The next chapter will seek to identify the Information Technology (IT) solutions that represent the most cost effective and efficient for meeting the DON's objectives. Existing architectures, technologies, and applications will be examined and analyzed. The advantages and disadvantages of existing solutions will be compared with those that have the potential to meet the mission objectives and budgetary goals. Finally, the advantages and disadvantages of developing mobile Native vs. mobile Web applications will be researched. This research will be conducted with the Navy Shore Command's mission and requirements in mind.

III. SURVEY OF ARCHITECTURES, TECHNOLOGIES, & APPLICATIONS

A. OVERVIEW

The introduction of the Personal Computer in 1974 facilitated a departure from the traditional main frame and the host client architecture as the accepted enterprise solution. Technological innovation and manufacturing economies of scale made the personal computer a cost-effective business solution. While PCs evolved into cost effective solutions capable of doubling output, the limiting factor remained the network bandwidth limitations and excessive server costs (Ceruzzi, 2003). File sharing applications grew larger and required more and more network bandwidth or server processing power. This led to the introduction of basic client server networks. Network bandwidth limitations were circumvented by sharing the application processing between the client and server, reducing the size of data packets transmitted across the network. The acceleration of technological innovations in server hardware and software allowed for the development of client-server multitier and client-server multitier distributed networks. These types of architecture enabled the use of extremely robust enterprise applications in spite of bandwidth limitations. Decentralization and specialization became increasingly paramount in maintaining productivity as the number of native applications increased exponentially.

The next phase of technological development brought about the information age and the Internet. The infrastructure that developed as a result allowed access to networks on a global scale. The wide spread development of client-server multitier architecture facilitated the next logical progression to client-server/web server architecture. Web servers allowed for access to applications and data from any location with access to the Internet. Bandwidth and network limitations were quickly becoming less of a concern in the development and deployment of applications. Web-based applications provided similar user experiences with exceptions limited to 3-D gaming and image processing (Charland & LeRoux, 2011). The Internet and proliferation of network access brought centralization and consolidation back to the forefront of IT solutions. Thin Clients are a viable option and can be combined utilizing the various client server architectures to provide significant cost savings and numerous advantages over Fat Clients. One of the primary metrics cited by comparative cost analysis studies was the Total Cost Of Ownership. This provided a general metric to guide organizations in implementing the optimal solution. The Client-Server Architecture employed also provided distinct advantages and disadvantages. The use of Native vs. Web-based applications must be considered when trying to decide how to provide the best user experience. Finally, security and network reliability may determine the best solution over cost and convenience.

B. TOTAL COST OF OWNERSHIP

1. Gartner Research Group TCO example

One of the primary disadvantages of moving to the Thin Client-Server Architecture is the costs associated with infrastructure, servers, and replacement of Fat Clients (Daniel, 1996). The Total Cost Of Ownership is an analytical model used to identify the difference between initial costs of acquisitions and the long term costs over a systems lifetime (Schmidt, 2011). TCO analysis has been used since the 1980's to help many organizations considering the acquisition of large enterprise IT systems identify the "hidden" costs of ownership. It can also be used to do direct comparison, such as in the case of thin clients vs. PCs (Schmidt, 2011). Table 1 is a TCO analysis of Thin-Clients vs. Fat-Clients by the Gartner Research Group. The assumptions made by the Gartner Research Group were:

- Windows Terminals (WT) were deployed using best practices. This includes the license cost for load balancing, resource management, and installations management services using Citrix MetaFrame XPe (Lowber, 2001).
- The budgeted number of users migrating is equal to the current number of active users, so the analysis includes 2500 MetaFrame XPe licenses (Lowber, 2001).

- The cost of WTs and PCs is assumed to be negligible assuming the cost of Windows Terminal Servers will offset the lower cost of the WTs when compared with the purchase of PCs (Lowber, 2001).

Direct costs are expenditures associated with hardware and software staffing and administration costs, hardware acquisition costs, and hardware maintenance costs. Indirect costs are expenditures for peer support, formal and informal training, as well as file and data management services. Migration costs apply only to Thin Clients and include client deployment, testing and optimizing applications, as well as training and implementation.

Table 1 shows a 32% TCO savings when comparing Thin Clients to Fat Unmanaged Clients (Lowber, 2001). The inclusion of Fat Managed Clients adds a new perspective on the Thin Client vs. Fat Client comparison. Fat Managed Clients refer to PCs considered locked down or configured by IT management so that user permissions are limited through the use of software driven management tools (Lowber, 2001). Table 1 shows this savings to be around 3.3% which is considerably less significant than PCs lacking an active asset management process. Including the cost of migration, the savings per month from switching from Fat Unmanaged Clients to Thin Clients allows a payback of just 3 months. The savings per month of Thin Clients when compared to Fat Managed Clients allows a payback period of 29 months. This is just short of two and a half years. Considering the longer life span of Thin Clients, this is still significant.

> The analysis was performed using Gartners Ti2 ("TI squared") software, with assumptions based on 2,500 desktops and 35 servers accessed by users from a central site and from two remote sites. The overall annual TCO is $12.9 million (or $5,160 per user) for thin clients (Windows terminal, or WTs), $13.4 million (or 5%, 360 per user) for "fat managed" Windows 2000 PCs, and $17.1 million (or $6,840 per user) for "fat unmanaged" Windows 2000 PCs. (Lowber, 2001 p.1)

TCO	Thin	Fat Managed	Fat Unmanaged
Direct	$5,276,197	$5,943,658	$6,710,772
Indirect	$5,478,388	$7,424,767	$10,402,545
Migration Costs	$2,176,204	0	0
Total TCO	$12,930,789	$13,368,426	$17,113,318
Payback Period		29 months	3 months

Table 1. TCO Results Thin vs. Fat (From Lowber, 2001)

Figure 12 shows that the inclusion of migration costs have the largest net effect on reducing the cost savings between Fat Managed Clients and Thin Clients (Lowber, 2001). Eliminating these costs through the use of a tailored implementation strategy and flexible framework should greatly reduce or negate these costs. It is also possible to use various current thin client solutions to allow IT management to incorporate recently replaced or upgraded PCs as thin clients until the end of their useful lifecycle. Replacing PCs already at the end of their useful life with Thin Clients will not only mitigate migration costs, but should further decrease the Thin Client TCO. The negation of migration costs increases the differential between both Fat Managed Clients and Fat Unmanaged Clients. Organizations deploying Thin Clients in lieu of PCs to multiple remote sites can further increase the TCO savings due to the decrease in hardware and software management and bandwidth limitations (Lowber, 2001).

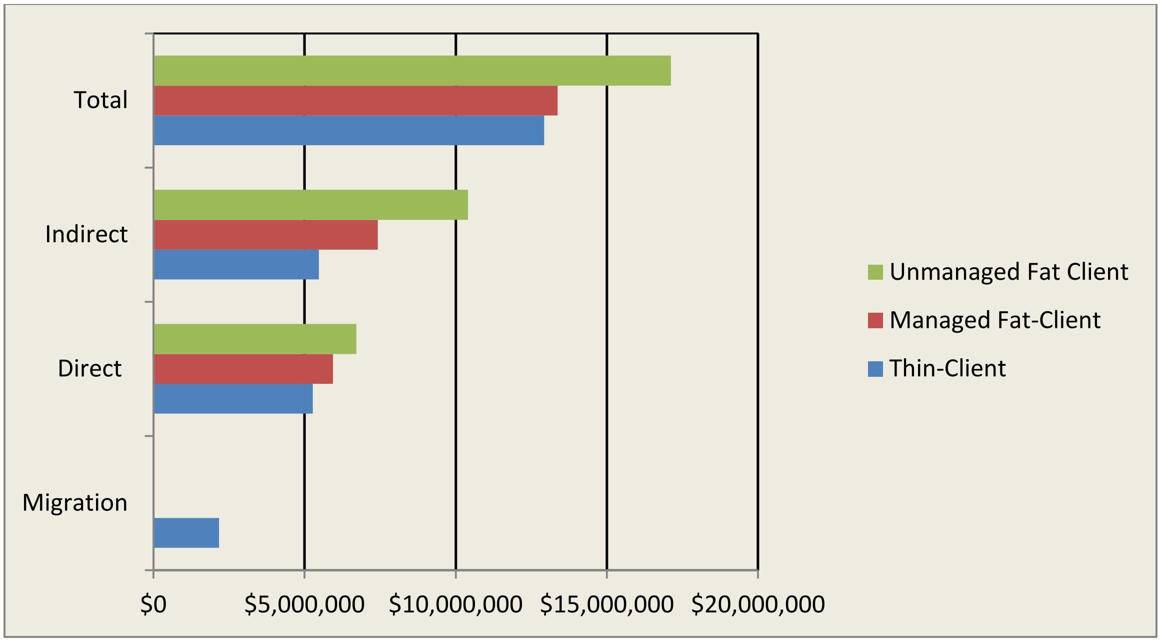


Figure 12. TCO Comparison: Thin vs. Fat (From Lowber, 2001)

2. Thin Client vs. Fat Client Energy Consumption

TCO takes several cost variables associated with owning and operating clients into account. Some of these costs can be implicit and difficult to measure or conduct direct comparisons. Energy consumption is one of the universal costs associated with the use of any type of client-server network (Anderson & Greenberg, n.d.). It is explicit and allows for direct comparison using the same metrics. The Wyse Technology Corporation commissioned a study to compare the energy consumption of Thin Clients vs. Fat Clients in a centralized client-server architecture. The thin terminals used in the study were: Wyse Winterm 3200LE Windows-based terminal, Wyse Winterm 3630LE Windows-based Terminal, and a Wyse Winterm 8230LE Windows custom-application terminal (Anderson & Greenberg, n.d.). The Fat Clients used were: a 1GHz desktop system with 128MB RAM and a 1.5GHz desktop system with 384MB RAM (Anderson & Greenberg, n.d.). Both Fat Clients were running Windows 2000. A Brand Electronics Model 21-1850 CI power meter was used to measure the devices while running applications from a terminal server using the ICA protocol (Anderson & Greenberg, n.d.). Figure 13 shows the test results by stage. These are the average power consumption of the thin clients without a monitor.

	3200	3630	8230
Plugged In	5 watts	8.27 watts	6.6 watts
Powered On	6.7	26.4	7.6
Running applications from ICA session	7.07	24.07	8
Logged out	7	23.8	8
Powered Down device	5.8	9.6	7.7

Figure 13. Average power usage among Wyse Winterm Thin Clients (From Anderson & Greenberg, n.d.)

Figure 14 compares the Wyse Thin-Clients (powered on) from Figure 13 against the power consumption of the Fat-Clients without the use of a monitor (Anderson & Greenberg, n.d.). A single PC without a monitor consumes around 90 watts of power (Anderson & Greenberg, n.d.). The Wyse 3630 has a built in display which could not be powered off separately for this test (Anderson & Greenberg, n.d.). Therefore, the 26.4 watts average power consumption from Figure 13 includes power consumed by the Liquid Crystal Display (LCD)

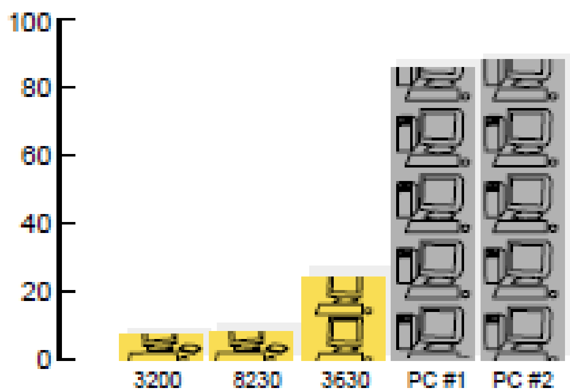


Figure 14. Power usage for thin client devices (without display) (From Anderson & Greenberg, n.d.)

Figure 15 graphically depicts the power consumption of the Wyse Thin Clients vs. the Fat Client desktops using a business class Cathode Ray Tube (CRT) desktop monitor connected to four of the clients. As stated earlier the Wyse 3630 has a built in LCD. The CRT consumed an average 85 watts of power (Anderson & Greenberg, n.d.). Power consumption for both Thin Clients and Fat Clients can be greatly reduced by using LCD displays vs. the CRT used in this study as evidenced by the built in LCD display of the 3630 (Anderson & Greenberg, n.d.).

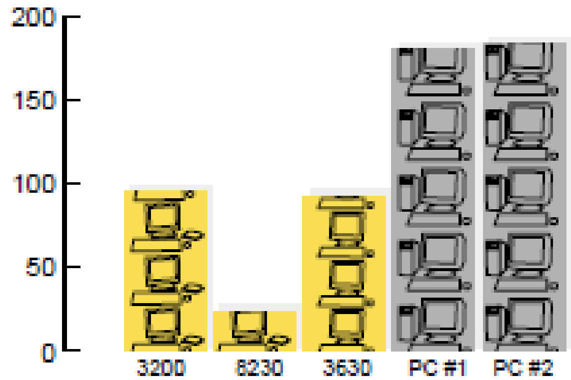


Figure 15. Power usage by thin client devices (with display) (From Anderson & Greenberg, n.d.)

The Wyse study then extrapolated the power consumption used per single unit to show the total power requirements for networks with 100 Clients, 1,000 Clients, and 5,000 clients. Figure 16 shows the totals by Client Device Type (Anderson & Greenberg, n.d.).

Client Device Type	Single Unit	100 Computers	1,000 Computers	5,000 Computers
3200	92 watts	9,200 watts	92,000 watts	460,000
3630	24 watts	2400 watts	24,000 watts	120,000 watts
8230	93 watts	9,300 watts	93,000 watts	465000 watts
PC	170 watts	17,000 watts	170,000 watts	850,000 watts

Figure 16. Power requirements for networks using thin client devices (with monitors) (From Anderson & Greenberg, n.d.)

The following formula allows organizations to estimate and compare the total cost of energy per year for their specific devices and energy rates (Anderson & Greenberg, n.d.).

N*P*H*kWh*52=Total Energy Cost Per Year

N = the number of desktop devices

P = the power (in kilowatts) used by each device

H = the number of hours each week that the devices are turned on

kWh=Cost per kilowatthour (“Electric Power Monthly,” 2011)

52 = the number of weeks in a year

C. THIN CLIENT-SERVER ARCHITECTURE

1. Advantages

a. Centralization

Moving to the Thin Client-Server Architecture provides consolidation of database centers and centralized application services (Machovec, 1997). In the Thin Client-Server Architecture, the servers do the data retrieval and processing. By moving all of the processing and application handling to the server, all of the data and software can be centralized. This provides universal access through centralized software management. Centralization also eliminates the duplication of application code associated with highly decentralized database distribution and individual or departmental development (Machovec, 1997).

b. Standardized Software

The centralization facilitated by Thin Client-Server Architecture allows for the standardization of software (Machovec, 1997). This is achieved by eliminating the numerous software versions on every Fat Client and consolidating them down to one version located on the server. Centralized software also allows IT departments to easily deploy software and application updates (Machovec, 1997). Each version of software and application only has to be updated on the server and is ready to be deployed to all users.

c. Increased Security and Data Integrity

Replacing Fat Clients with Thin Clients increases security and control at the desktop and network level (Machovec, 1997). Thin Clients have no local storage or peripheral media devices for the user to install third party software or inadvertently introduce viruses and malware. Access control is set on the server and cannot be accessed from the Thin Client eliminating security gaps. Fat clients require the user or IT staff to periodically update and back up local data (Machovec, 1997). Centralized servers and Thin Clients eliminate reliance on the user and shift control back to IT departments. This ensures data is regularly and efficiently backed up using server-based backup capabilities. Thin Client-Server Architecture also enables the use of centralized file sharing via the network (Machovec, 1997).

d. Reduced Administration and Troubleshooting Time

The Thin Client-Server solution reduces administration and troubleshooting expenses through standardization (Machovec, 1997). Centralized servers and databases allow IT departments to manage all data and applications from one central hub. Troubleshooting software or application issues are confined to the server where they reside. The use of single version software and operating systems on centralized servers also reduces the need for user training associated with individual Fat Client replacements or upgrades (Machovec, 1997). IT departments can easily monitor and track required upgrades and the potential need for organizational wide training due to a single mass software or application rollout.

e. Reduced Costs and Equipment Disparities

The use of Thin Clients reduces space and depending on the type of Fat Client replaced, reduces energy requirements by 14% (David, 2003). The larger the organization and Client deployment, the greater the costs savings due to the reduced energy consumption. Thin Client solutions are multifaceted and provide solutions that allow organizations to use existing Tubby Clients or Low Spec PCs to reduce the cost of migration to the Thin Client-Server Architecture. Thin Client-Server Architecture addresses equipment and application disparities that may exist in an organization due to departmental budgetary constraints (Machovec, 1997). Larger departments in an organization may garner more of a share of the organizational budget allowing them to purchase IT equipment more often. This leads to disparities in equipment and added infrastructure complexity. Since Thin Clients use software and applications on the centralized servers, everyone in the organization has access to the most relevant organizational content.

2. Disadvantages

a. Infrastructure and Migration Costs

Switching to the Thin Client-Server Architecture can add significant upfront costs due to consolidation of database systems and the restructuring of IT infrastructure needed to take full advantage of centralized services (Machovec, 1997). Legacy applications may require separate migration or the use of middleware to function properly with a Thin Client solution. This increases the cost of migration and the complexity of the Architecture. Thin Client-Server solutions shift the burden to the server and may require organizations to purchase more robust servers to ensure all users can be accommodated. The network infrastructure must be able to support bandwidth and reliability requirements. The closer to 100% network uptime required, the more costly the infrastructure upgrades (Machovec, 1997).

b. Network Dependency and Lack Of User Requirements

Moving to centralized software and applications requires network access in order to perform tasks. If the network goes down or users are unable to access server resources, pure Thin Client solutions offer no alternative ability to run applications. This creates over reliance on network connectivity (Machovec, 1997). Organizations that decide to reduce costs by eliminating redundant servers also risk the possibility of a single point of failure (Machovec, 1997). Any server that exists as the single source of an application or data can be a potential single point of failure. If the server goes down, users have no alternative means of access. Organizations that do not employ network solutions robust enough to support Thin Client-Server applications may experience lost productivity due to network latency (Machovec, 1997). It is vital that the entire IT infrastructure be considered when implementing Thin Client solutions.

c. User Bias and Lack of Advanced Application Support

Users accustomed to the localized control and structure of Fat Client-based architectures may be resistant to accept Thin Client solutions (Machovec, 1997). It is up to IT departments and organizations to ensure users are informed and knowledgeable on what migration to Thin Clients means to them. Organizations must be careful to ensure the user still has the largest voice in development or acquisition of applications to ensure that IT departments are not instituting a one size fits all environment (Machovec, 1997). Organizations can garner more user buy in if the users feel they have a voice during the process. Including end users also ensures any graphics or multimedia applications that are not supported by a Thin Client solution are identified and addressed upfront and early (Machovec, 1997). These users may require alternate solutions such as high powered PCs used in conjunction with thin clients. If enough users are identified, it may be in the best interest of the organization to seek alternative architectures to increase efficiencies.

D. NATIVE APPLICATIONS

1. Advantages

a. System Costs

Native Applications or Client-Server Applications are developed to suit specific user needs and are generally optimized for use on a specific system. Software and application providers write the application to meet user requirements and specifications; this means the final product must meet this criteria before the vendor is paid in full. One of the biggest advantages of this is that the organization is in control of when upgrades are needed and only pays for those features as required (Clouse, n.d.). Depending on the contract, the organization can sever ties with the software provider if support is no longer needed or the decision to move to a different vendor is made. In the case of outsourced services, the organization can continue to use the application since it is hosted internally on the local network.

b. Environment

Native Applications remain on the locally hosted network giving internal staff full control of the environment (Clouse, n.d.). All decisions regarding IT solutions are controlled by the organization. The software vendor only has access to IT systems if permission is granted by the organization.

c. IT Support

Internal staff provide the support and decide when system upgrades, user seat software, and database migrations occur (Clouse, n.d.). The Application Service Provider has to work with IT support staff and schedule according to what they deem is best for the organization.

d. User Seats

Native Applications take advantage of a Fat Clients processing power and local storage capacity. Graphic and Media intensive applications run faster due to the shared processing and local storage provided by the user's Fat Client (Clouse, n.d.). Each user has his or her own seat and can configure their applications to best suite his or her own needs and preferences.

e. System Access

Most Native Applications do not require network access to function. Software is only accessible through the local network, increasing the security of the application and data (Clouse, n.d.). Users can initiate and use the application until network access is available to upload data to the database if needed.

2. Disadvantages

a. System Costs

Initial cost of application development can be high and is usually paid for up-front (Clouse, n.d.). If user requirements and specifications are not captured accurately, the application delivered may not meet user needs and still meet contract requirements. Organizations can end up paying for a product that does not work. Software and application vendors may require service contracts that cost a certain percentage of the entire system cost per year (Clouse, n.d.). Organizations can end up paying for these services even if they are not used.

b. Environment

Organizations bear all costs associated with purchasing, maintaining, and upgrading network and IT structure (Clouse, n.d.). The use of Native Applications is a commitment to maintaining your own network services in lieu of outsourcing. This may conflict with the organizations core competencies and detract from day to day operations.

c. IT Support

Internal IT staff must decide what upgrades the organizations needs and how best to deploy them. Vendors may try to push multiple upgrades and features at added expense (Clouse, n.d.). Issues that arise with upgrades must be dealt with in house or via vendor support. If vendor support is not local, this can require granting vendor IT support staff remote access making it even harder to diagnose.

d. User Seats

Every user that requires access to the application is charged for a seat or license fee (Clouse, n.d.). Organizations may end up paying for users who require access, but rarely use the application. This leads to inflated costs relative to application use and productivity.

e. System Access

These types of applications can only be used on devices capable of processing and local storage. This negates the use of most thin client solutions. Users have to be on the local network to access applications limiting their ability to Telework or access data while traveling on business. Organizations wishing to grant users remote access must invest in separate application software such as Citrix or PC Anywhere (Clouse, n.d.). These applications also require additional user software. Opening up portals for remote access into the local network decreases the security provided by using Native Applications.

E. WEB APPLICATIONS

1. Advantages

a. System Costs

Web services can be outsourced allowing organizations to subscribe just to the services and applications that they need (Clouse, n.d.). Organizations that outsource or host their own Web Applications can take advantage of Thin Client-Server solutions. The TCO of these systems has been shown to be significantly less vs. traditional Fat Client solutions.

b. Environment

Outsourced Web Applications include the hardware, software, and database systems in the price of the application service (Clouse, n.d.). No local network service upgrades are required to take advantage of outsourced Web Applications. Organizations that host their own Web Applications have more control over the environment than those using Native Applications. This provides IT staff with a centralized environment that they can easily administer.

c. IT Support

Outsourced Web Applications eliminate internal IT staff responsibilities for system upgrades, user seat software and database migrations, and server maintenance

(Clouse, n.d.). The vendor is responsible for administering all facets of Web Application support. Organizations hosting their own Web Applications are able to take advantage of centralized Web Server administration. Internal IT staff only need to update one Web Application version on the server vs. every user seat.

d. User Seats

User seats are actually application sessions on the Web Server initiated by the user. The software exists on the Web Server instead of individual user Fat Clients (Clouse, n.d.). This allows multiple users to initiate Web Server sessions at one time using a single software version. Organizations outsourcing Web Applications only pay for the access they need rather than individual User Seats that may or may not be fully utilized. Web Applications allow organizations hosting their own services significant flexibility to grow user seats as they are needed. Adding a new user seat is as simple as providing the user the correct logon credentials and web address.

e. System Access

Web applications are accessible using a standard web browser and an Internet connection. These applications work well in a Fat Client environment and are the preferred application solution for Thin Client Architectures. Users can access Web Applications from any location with Internet Access using any device with a web browser. Organizations using Web Applications have remote access inherently built in due to the Web Enabled nature of the applications.

2. Disadvantages

a. System Costs

Web Applications that are outsourced usually require subscriptions paid by organizations on a yearly basis in order to maintain access to services (Clouse, n.d.). If the organization decides to sever the relationship or move to a different service, access to the application and data is also severed. The cost of lost data and productivity must be considered when outsourcing Web Applications. Organizations that host their own web services are not subject to loss of access, but incur all costs associated with building and maintaining Web services in addition to normal operating costs.

b. Environment

Outsourced Web Applications limit the organizations access to the system and internal IT staffs control of the environment. The organization must rely on the Application Service to provide the required level of security and access control. This requires explicit communication between the organization and Web Application Service.

c. IT Support

Organizations outsourcing Web Applications have no direct control or access to the system and therefore no say in how the network is managed (Clouse, n.d.). Any issues must be addressed by the service's IT support staff. Application Service Providers may charge additional fees to give an organization priority access to network administrators when issues arise. Troubleshooting problems can become an issue when the Web Application Services staff and the organizations staff are unable to narrow down the point of failure. This can lead to organizational conflicts where neither party is willing to accept the responsibility for application or network issues.

d. User Seats

Users do not have local access to the application or data and must be connected to the Internet to access applications (Clouse, n.d.). No network access translates to lost productivity and frustrated users.

e. System Access

Vulnerabilities in the Web Application Services network are inherited by organizations outsourcing Web Applications. Web Applications, whether outsourced or hosted internally are subject to outside attacks via the Internet. The remote access that makes these applications attractive also allows attackers access to the applications and potentially the data.

F. SMART MOBILE WIRELESS DEVICE APPLICATIONS

1. Mobile Native Applications

a. Advantages

Native Applications are written using compiled code native to the operating system. This makes some mobile applications faster than their Web-based counterparts. Since Native Applications use code common to the operating system and the device, these applications provide excellent common user-interface controls and experiences (Charland & LeRoux, 2011). Native Applications provide better application control and scrolling than Web Applications. This provides the user with better input feel and easier navigation. These Applications can be accessed and provide functionality even when the cellular network or Wi-Fi is unavailable.

b. Disadvantages

Native Applications written to work on multiple mobile devices require developers with a myriad of programming skill sets (Charland & LeRoux, 2011). Figure 17 represents all of the skill sets required to develop Native Applications for all of the major Smart Wireless Mobile Devices currently in use.

Mobile OS Type	Skill Set Require
Apple iOS	C, Objective C
Google Android	Java (Harmony flavored, Dalvik VM)
RIM BlackBerry	Java (J2ME flavored)
Symbian	C, C++, Python, HTML/CSS/JS
Windows Mobile	.NET
Window 7 Phone	.NET
HP Palm webOS	HTML/CSS/JS
MeeGo	C, C++, HTML/CSS/JS
Samsung bada	C++

Figure 17. Required Skill Sets for Nine Mobile OSes (From Charland & LeRoux, 2011)

This chart shows how expensive and complicated Native Application development can be, whether developed in house or outsourced. This coupled with the numerous differences in platform software development kits (SDKs) makes standardization near impossible (Charland & LeRoux, 2011). These Applications tend to outperform mobile Web Applications, but they typically take much longer to develop and download. They also require local device storage and dedicated processing. On a typical Fat Client this does not become a major issue. However, most mobile devices have limited storage space, processing capabilities, and battery life that can adversely affect the performance of Native Applications.

2. Mobile Web Applications

a. Advantages

Mobile Web Applications are developed using an interpreted language such as JavaScript(JS). JavaScript and other interpreted languages take advantage of browsers that are common to the mobile operating systems and allow access via native code. Developers can use the browser to call the JavaScript interface via the native code (Charland & LeRoux, 2011). The Webview is then used to call native code from the JavaScript (Charland & LeRoux, 2011). This “PhoneGap” technique enables developers to create Web Applications using HTML, CSS, and JavaScript that take advantage of native device feature sensors via the JS API (Charland & LeRoux, 2011). Now the Web Application is able to use device features that were previously only accessible by Native Applications. Some of these applications also provide longer battery life than their Native Application counterparts (Charland & LeRoux, 2011).

Mobile frameworks and support for next generation mobile browsers are quickly being added that will facilitate the expansion of APIs for the browser in the near future. The World Wide Web Consortium has put together a Device APIs Working Group to “create client-side APIs that enable the development of Web Applications and Web Widgets that interact with devices services such as Calendar, Contacts, Camera, etc” (*Device APIs working group - W3C*). These efforts are quickly shoring the gap between

Native Application and Web Application features. Web Applications require developers to write far less code in one code base for multiple devices and operating systems (Charland & LeRoux, 2011). This significantly reduces costs, maintenance, and makes troubleshooting and support easier.

b. Disadvantages

Web Applications have a limited number of SDKs or built in controls available to developers. The lack of development tools currently available forces developers to create their own controls to facilitate browser to user interface code compatibility. Creating these controls and SDKs is time consuming, costly, and lacks standardization. The lack of SDKs and built in controls limits Web Applications capacity to provide the same user interface experience as Native Applications (Charland & LeRoux, 2011). Web Applications use interpreted code which can decrease performance and cause network latency. The bigger and more complex the Web Application, the more code there is to interpret. Execution times increase as the amount of interpreted code grows. Part of the user experience is navigating through an application. Navigation includes the user's ability to scroll through application pages. Web Technology is not yet able to reproduce the native device scrolling features offered by Native Applications.

G. SUMMARY

The TCO of Thin Clients, including the cost to migrate, may significantly reduce the overall cost associated with IT solutions. Thin Clients also represent a significant savings in energy consumption, which can be directly compared to Fat Clients. The reduced energy consumption and yearly power costs save the organization money and support green initiatives. The advantages and disadvantages of Thin Client-Server architecture were analyzed. Native and Web-based applications were researched using systems cost, environment, IT support, user seat, and system access to compare the advantages and disadvantages. Finally, mobile Native and mobile Web-based applications were researched and compared.

Chapter IV provides a detailed cost comparison using TCO as a guideline. The cost data used was provided by CNRFC and representative of Naval Shore Installations. Current DoD/DON initiatives and examples of Native and Web Applications will be provided and analyzed. Finally, a student developed Web-based application is provided and discussed. The analysis in Chapter IV is in support of the research done in preceding chapters and will be used to make final conclusions and recommendations to the reader.

IV. COST COMPARISON, CURRENT INITIATIVES & EXAMPLES, STUDENT DEVELOPED WEB APPLICATION

A. COST COMPARISON

The DoD and DON are increasingly under budgetary constraints and pressure to do more with less. The DDCIO (N) has directed commands to reduce IT systems costs by seeking increased efficiencies and less expensive solutions (Dorsett, 2011). The replacement of Fat Clients with Thin Clients in the existing Navy Marine Corps Infrastructure has been identified as a possible solution. Thin Clients shift the majority of application processing to the Server facilitating an increase in server utilization rates as mandated by the DDCIO (N) (Dorsett, 2011). Commander Navy Reserve Forces Command is a shore based Echelon II command directly tasked with implementation of the DDCIO (N)'s IT initiatives. CNRFC is representative of the DON's major shore commands and outsources IT services to NMCI. The DON NMCI services contract ended 30 September 2010 (Sternstein, 2010). Hewlett Packard Enterprise Services was awarded a Continuity Of Services Contract (COSC) after the expiration of the original NMCI services contract. The COSC contract allows the DON time to migrate NMCI to the Next Generation Enterprise Network (NGEN). The contract is a fixed price, base contract with options through 2015 (Sternstein, 2010). The COSC allows the DON the ability to seek competitive bids from other vendors outside of HP Enterprise Services for NGEN. The following NMCI cost data was provided by LCDR Byron Moss, CNRFC N63, and will be used to provide the current NMCI baseline for comparison to Industry standard costs.

1. CNRFC NMCI Fat Client Costs

This section outlines the assumptions made in creating Table 2 from the data provided by CNRFC staff. All personnel will require an average desktop PC. Power Users and Developers will then customize their desktops using hardware and software components from Table 3. Some personnel require the use of a laptop in addition to their standard desktop PC in order to facilitate mobile access to NMCI services while still adhering to security protocol and NMCI standards. Special Software is considered to be any Native Application not included in the NMCI standard application suite. These applications are too numerous to list and include Native legacy applications migrated over to NMCI. The monthly cost of special software shown is for Adobe Acrobat Professional, one of the more prevalent Commercial off-the-Shelf (COTS) Applications required by advanced users. The cost associated with Adobe Acrobat Reader Professional is representative of the cost for the average Native Application considered Special Software.

Monthly	**Desktops**	**Laptops**
Per Seat	$102.55	$110.55
Special Software (Variable)	$6.12	$6.12
Drop (LAN Connection)	$23.85	$23.85
Total (Monthly)	132.52	140.52
Total (Yearly)	$1590.24	$1686.24

Table 2. CNRFC NMCI Fat Client Monthly Costs (B. D. Moss/CNRFC N63, Personal communication, September 9, 2011)

This data shows the current monthly COSC charges for NMCI desktops and laptops (Fat Clients) currently in use at CNRFC. The monthly charge per seat for desktop PCs is $102.55 and $110.55 for Laptops. Special Software is variable and applications dependent. The example chosen costs $6.12 a month for both desktop PCs and Laptops. The monthly charge per Local Area Network (LAN) drop is $23.85 for both desktop PCs and laptops. The LAN drop is the wired connection required for NMCI network access. The total yearly cost for a desktop, including a single native application (special software), is $1590.24. The total cost for a laptop is slightly higher at $1686.24.

This section outlines Table 3 assumptions. The onetime fee for special software is for installation of Adobe Acrobat Reader Professional. This Native application is representative of those most often required outside of the NMCI standard suite of preloaded applications. Outlook Web Access is included as part of the standard NMCI monthly charge per seat from Table 2. The price for a LAN drop includes wiring to the switch and installation of an Ethernet connection. Every user requires one drop per NMCI seat. The costs for peripherals and miscellaneous hardware purchased through NMCI varies based on user requirements. An average price was calculated using the lowest and highest costs associated with current offerings per the NMCI Contract Line Item Number (CLIN) catalog for standard equipment.

One Time Fees	**Desktops/Laptops**
Special Software (Variable)	$228.00
Outlook Web Access	Included
Drop (LAN Connection)	$659.53
PRICES VARY BASED ON USER REQUIREMENTS	**AVERAGE PRICE**
Printer ($205-$275)	$240.00
Keyboard ($25-$60)	$42.50
Mouse ($22-$40)	$31.00
Monitors ($192-$379)	$285.50
External Hard Drive ($150-$159)	$154.50
Computer RAM ($116-$260)	$188.00
Total	$1829.03

Table 3. CNRFC NMCI Fat Client One Time Fees (B. D. Moss/CNRFC N63, Personal communication, September 9, 2011)

The data provided for peripherals and miscellaneous hardware are one time fees and assessed as part of the total cost per user seat during the first year. Special Software costs are $228.00 per Native Application installed outside of standard software included. Outlook Web Access is included. Network access is provided via LAN Ethernet drops at

$659.53 per drop. The average cost per NMCI printer is $240.00. Keyboard and mouse on average cost $73.50 combined and are generally accepted as required peripherals. Monitors are also considered required peripherals. LCD monitors represent the standard equipment at an average cost of $285.50. External Hard Drives are typically reserved for power users and purchased as required. The average charge for Random Access Memory is $188.00 and varies based on the type and amount required.

2. CNRFC Fat Client vs. Thin Client Comparison

This section lists the assumptions used in calculating Table 4 data. The Gartner Research group used 2500 users to calculate the TCO for Thin Clients, Fat Managed Clients, and Fat Unmanaged Clients. Table 4 calculations used the CNRFC data and extrapolated to account for the 2500 user baseline. NMCI desktop costs for a standard user were calculated using Table 4 monthly costs per desktop and per drop. The one time fees from Table 3 used were drop, printer, keyboard, mouse, and LCD monitor. The 2500 user baseline was modified to account for 250 power users or 10% of the standard 2500 baseline. These users represent those requiring Special Software, External Hard Drives, RAM upgrades, and a Laptop. The overall TCO savings from Chapter III were then used to find the potential overall yearly savings from migrating to a Thin Client solution. The cost of migrating to Thin Clients was included in the TCO savings calculations and may be negated depending on the implementation strategy chosen. It should be noted that this assumes the cost savings realized by the provider will be passed on to the DON if thin clients are incorporated as part of the NGEN transition.

Currently CNRFC outsources IT services on a COSC basis to HP Enterprise Services. The data provided by CNRFC was used to calculate costs in Tables 2 and 3. Those costs were used to calculate Table 4 data. The total cost per standard user seat is $2,775 per year. Laptops are provided to power users in addition to their desktop pc for remote access or mobility purposes and not applicable in this row. The total cost per power user is $3,419 a year for desktops and $2,605 a year for laptops. Using the $2,775 per standard seat and extrapolating out using the 2500 seat baseline minus the 10% gives us $6,244,493 for 2250 standard NMCI seats. The total cost per year for 250 power users is $854,818 for desktops and $651,193 for Laptops. Adding the total cost per year for standard seats to the total cost per year for power users equals $7,099,310 per year for NMCI desktops. The total yearly cost for all 2500 users is $7,750,503, which is the total cost for 2250 standard seats and 250 power users plus laptops. The product of the TCO savings comparing Fat Managed Clients vs. Thin Clients from the Gartner study was then multiplied by the total yearly cost to get a possible savings of 3.3%. This procedure was then applied to the total yearly cost using the TCO savings comparing Fat Unmanaged Costs vs. Thin Clients of 32%. The average TCO savings was then calculated using Fat Managed and Unmanaged costs. The average was 17.65% or $6,382,539.

Yearly	**Desktops**	**Laptops**
Total Cost Per Standard Seat	$2,775	N/A
Total Cost Per Power User	$3,419	$2,605
2250 Standard NMCI seats	$6,244,493	N/A
250 (10% of 2500- Power Users)	$854,818	$651,193
Total Cost	$7,099,310	$651,193
Total Yearly Cost (All 2500 users)	$7,750,503	
Total Yearly Cost (With 3.3% TCO savings)	$7,494,736	
Total Yearly Cost (With 32% TCO savings)	$5,270,342	
Average Cost Savings (Average of TCO savings)	$6,382,539	

Table 4. CNRFC NMCI Data Extrapolated Using Table1 As A Costing Model

Figure 18 compares the total yearly cost of the current NMCI solution at CNRFC extrapolated for 2500 users. This data was then used in conjunction with the TCO savings calculated in Table 1 to find the potential cost savings of migrating to Thin Clients from a NMCI Fat Client Managed and Unmanaged solution in Table 4. The average cost savings for Figure 18 was calculated based on insights gained from the Gartner Research Groups Thin Client case study. The TCO calculated for Fat Managed and Unmanaged Clients made several assumptions about the PCs configuration and IT management practices. The average of the TCO savings was calculated to adjust for these assumptions and provide a more moderate cost estimation based on average IT management practices. Figure 18 graphically illustrates the cost savings comparison.

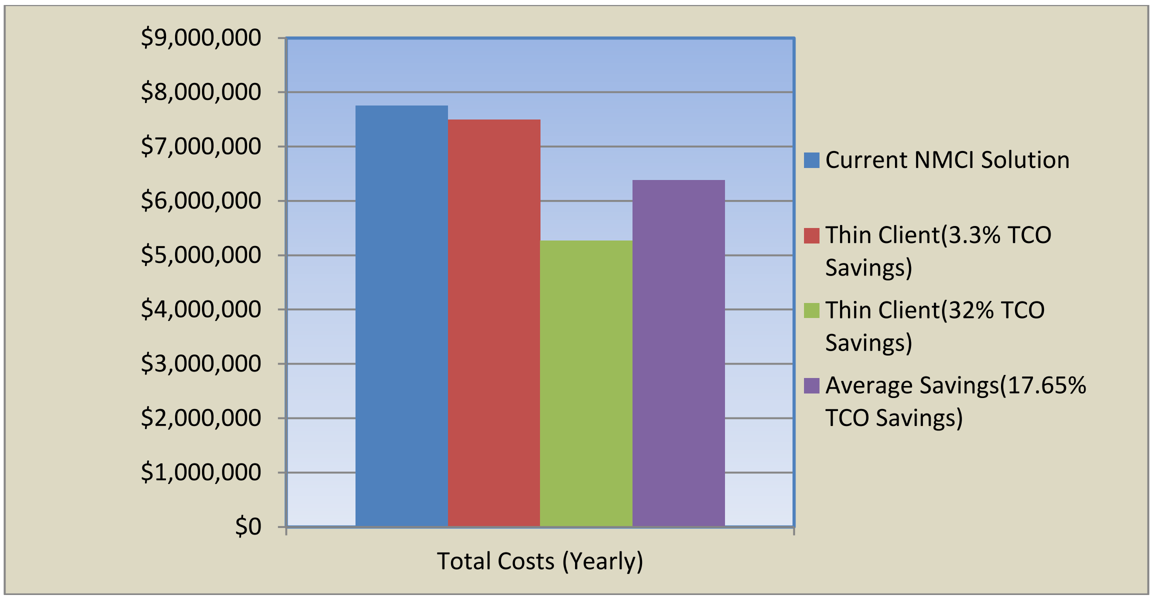


Figure 18. CRNFC Current Fat Client Solution and Thin Client Comparison

3. Power Consumption and Costs

This section details the assumptions made for power consumption calculations and costs. The power consumption of a standard 17 inch Dell LCD was used for all Fat Client and Thin Client calculations. The NMCI standard PC's power consumption is represented by an average Dell Desktop Computer. The TCO analysis from Table 1 does not include the cost of energy as part of the overall calculation. The analysis in Table 5 will calculate the cost of energy using NMCI standard components and Wyse Thin Clients.

The monitor used with all clients to calculate power consumption costs was the Dell E170S 17 Inch LCD (DELL, 2011). The Dell LCD used an average of 17 Watts of power. The Dell Optiplex 380 DT with Intel Pentium Dual Core E5300/2.60GHz, 2M cache, and 800MHz Front Side Bus (FSB) was used for the Fat Client NMCI baseline power consumption (DELL, n.d.). The Dell Optiplex 380 uses an average of 91.5 W according to the EPA standard. The Wyse P20 Thin Client has a Teradici 1100P PCoIP chip and a 128MB RAM (K. Heydler/Wyse Federal Insides Sales Rep, Personal communication, September 14, 2011). The Wyse P20 average power consumption of 15.36 Watts was measured with peripherals connected. The Wyse Xenith Thin Client has a 1GHz Via C7 Processor, 128MB Flash RAM, and 512 MB system RAM (K. Heydler/Wyse Federal Insides Sales Rep, Personal communication, September 14, 2011). The Wyse Xenith's average power consumption is 7 Watts or less measured with peripherals connected. The Wyse Xenith Pro has a 1.5GHz AMD Sempron Processor, 128 MB Flash RAM, and 512 MB RAM (K. Heydler/Wyse Federal Insides Sales Rep, Personal communication, September 14, 2011). The Wyse Xenith Pro's average power consumption is 14.6 Watts measure with peripherals connected. Table 5 lists the power consumption associated with each device.

Dell E170S 17 Inch LCD	**Dell Optiplex 380 DT**	**Wyse P20**	**Wyse Xenith**	**Wyse Xenith Pro**
25 W (Max)	PPFC 105W	----------	----------	----------
17 W (Typical)	EPA 91.5W	15.36 W Average	7 W or Less Average	14.6 W Average

Table 5. NMCI Standard Equipment And Wyse Thin Client Power Consumption

Table 6 shows the energy cost per year for the NMCI standard Fat Client and several potential Wyse Thin Client solutions. All energy calculations were done using the power consumption of the Dell E170S 17 Inch LCD monitor plus the device. The Average Retail Energy Price for commercial entities up to and including May 2011 was

.10cents/kWh ("Electric Power Monthly," 2011). This was used to calculate the total price per year in each column. The following formula from Chapter III was used ("Electric Power Monthly," 2011):

N*P*H*kWh*52=Total Energy Cost Per Year

N = the number of desktop devices

P = the power (in kilowatts) used by each device

H = the number of hours each week that the devices are turned on

kWh=Cost per kilowatt-hour ("Electric Power Monthly," 2011)

52 = the number of weeks in a year

The number of hours each week that the devices are turned on or variable "H" from the above formula was assumed to be an industry standard of 40 hours/week. It was assumed that the devices would be shut down or enter standby mode outside of normal business hours. Power consumption of the devices in standby was assumed to be negligible compared to peak/average power consumption during business hours.

Number Of Devices	**100**	**500**	**1000**	**2500**
Dell Optiplex 380 DT	$2,272.60	$11,362.99	$22,725.98	$56,814.94
Wyse P20	$677.80	$3,389.00	$6,778.00	$16,944.99
Wyse Xenith	$502.69	$2,513.47	$5,026.94	$12,567.36
Wyse Xenith Pro	$661.88	$3,309.40	$6,618.81	$16,547.02

Table 6. Energy Cost Per Year

Figure 19 graphically illustrates the potential cost savings and energy efficiencies that can be achieved by migrating to Thin Client Solutions. The average energy costs savings of the three Wyse Thin Clients was calculated and used to calculate the percentage savings compared to the representative NMCI Fat Client. The average energy costs savings is 37.3%. The energy savings represented in column one do not appear that drastic. However, extrapolating out using 2500 PCs and Thin Clients provides an enterprise wide migration energy cost picture that is significant.

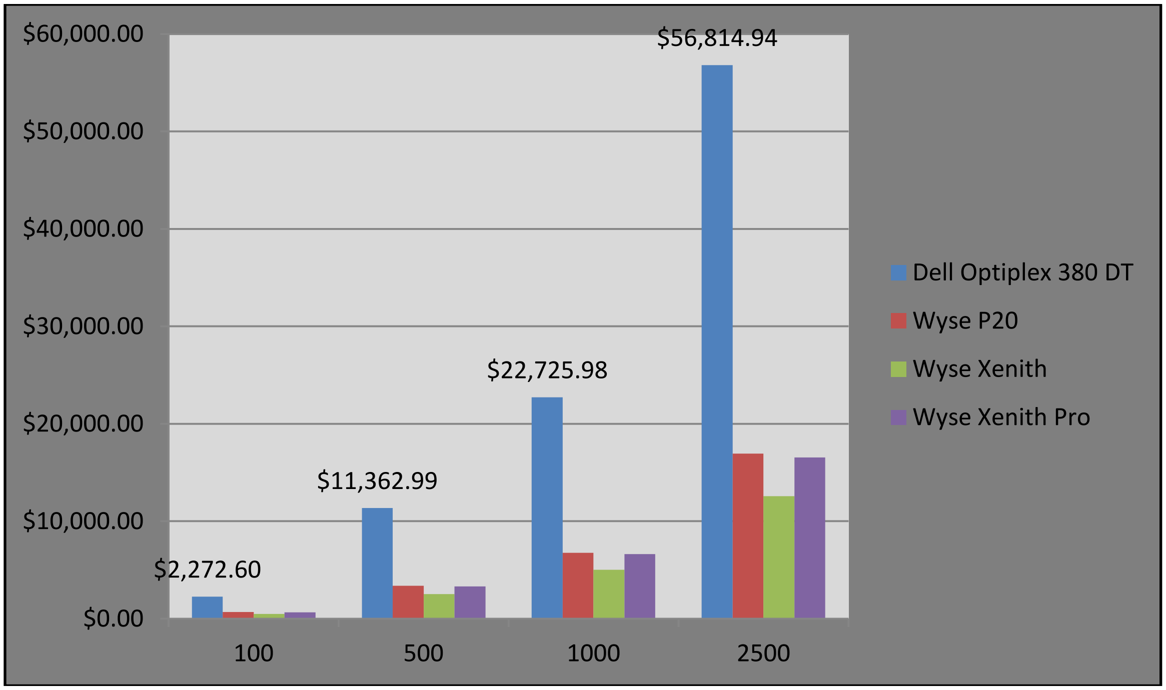


Figure 19. Graph Of Energy Cost Per Year

B. CURRENT INITIATIVES AND EXAMPLES

The DON has already taken steps to increase the usability and access of applications for active duty and reserve personnel. Web Based applications allow Active duty and Reserve personnel the ability to monitor pay and allowances and administer their own service records. The Reservists have web applications available through SPAWAR to meet their specific needs. CRNFC has taken this a bit further and offers a command specific web portal for access to multiple web based applications for staff and assigned personnel. Mobile Native Applications for DoD personnel have also been developed to take advantage of the mobile device optimization and offline functionality.

These target service members with access to Smart Mobile Devices and specific user requirements. This section will address these current initiatives and provide examples of application functionality and access.

1. myPay

Figure 20 shows the Secure Web Login Screen allowing Service members access to myPay web application. The web address for myPay is https://mypay.dfas.mil/mypay.aspx (DFAS, n.d.). This Web Application is accessible from any client with a web browser including Smart Mobile Wireless Devices. If accessing the myPay Web Application using a Smart Mobile Wireless Device, the application will automatically detect the mobile operating system used and optimize for that OS (DFAS, n.d.). The service member creates an account via the "Create an Account" tab to access the Web Application using a LoginID and Password. The myPay web application is also accessible using a Common Access Card (CAC) and Reader. Figure 20 shows a screen shot of the myPay web portal and secure login screen.

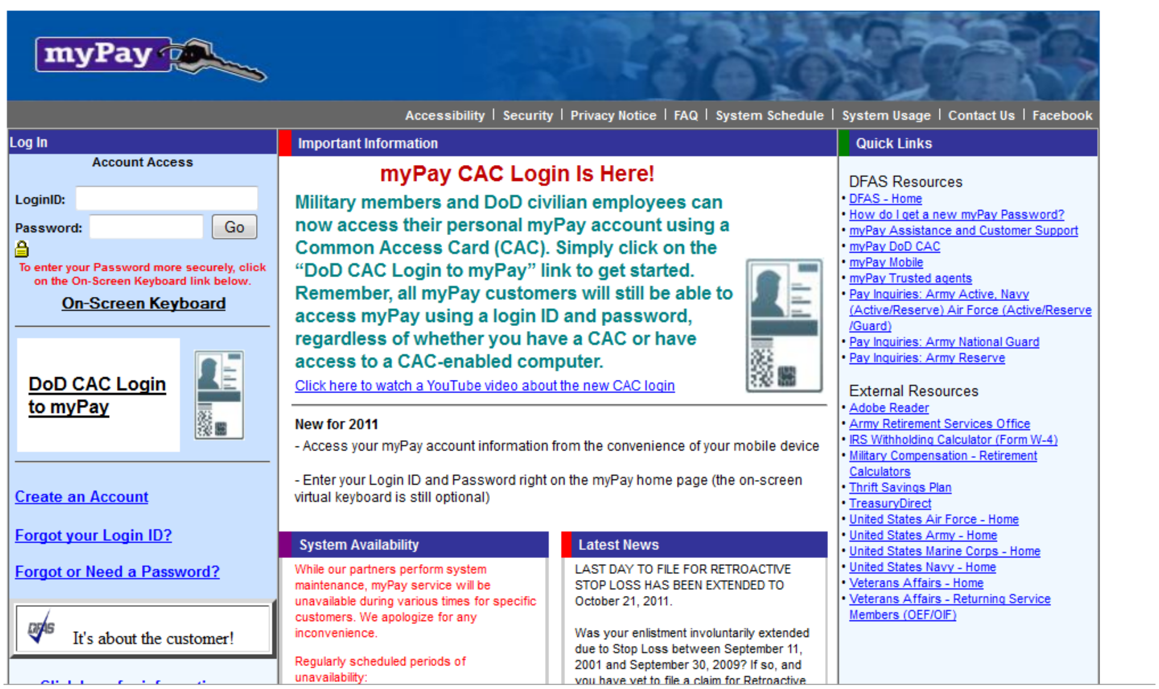


Figure 20. myPay Secure Web Login Screen (From DFAS, n.d.)

Once the user has authenticated via the LoginID and Password or CAC, they are granted access to the main menu. The main menu allows users to view pay statements, make pay changes, and view or verify tax information. The Web Application gives the user the ability to administer his or her own pay and allowances from any Internet accessible location using multiple different device types. Figure 21 is a screenshot of the main menu screen and shows the various user accessible functions.

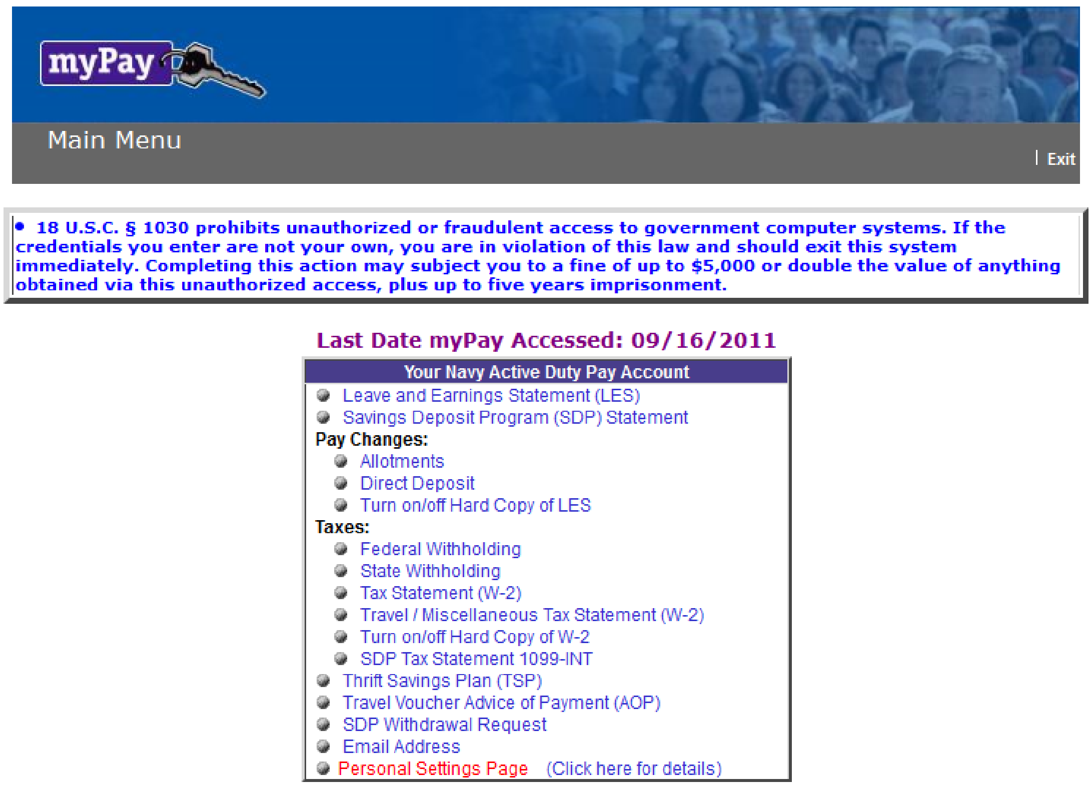


Figure 21. myPay Main Menu Screen (From DFAS, n.d.)

2. Navy Standard Integrated Personnel System (NSIPS) and Electronic Service Record (ESR)

NSIPS gives service members access to their entire electronic service record and provides the ability to make changes. NSIPS is can only be accessed by a Client device with CAC authentication capability. The Web Application is accessed via a web browser using a secure web browser connection. The logon screen then uses certificates pulled from a user CAC and the user provided password to grant access to the ESR main menu. Figure 22 shows the NSIPS Web Application login screen.

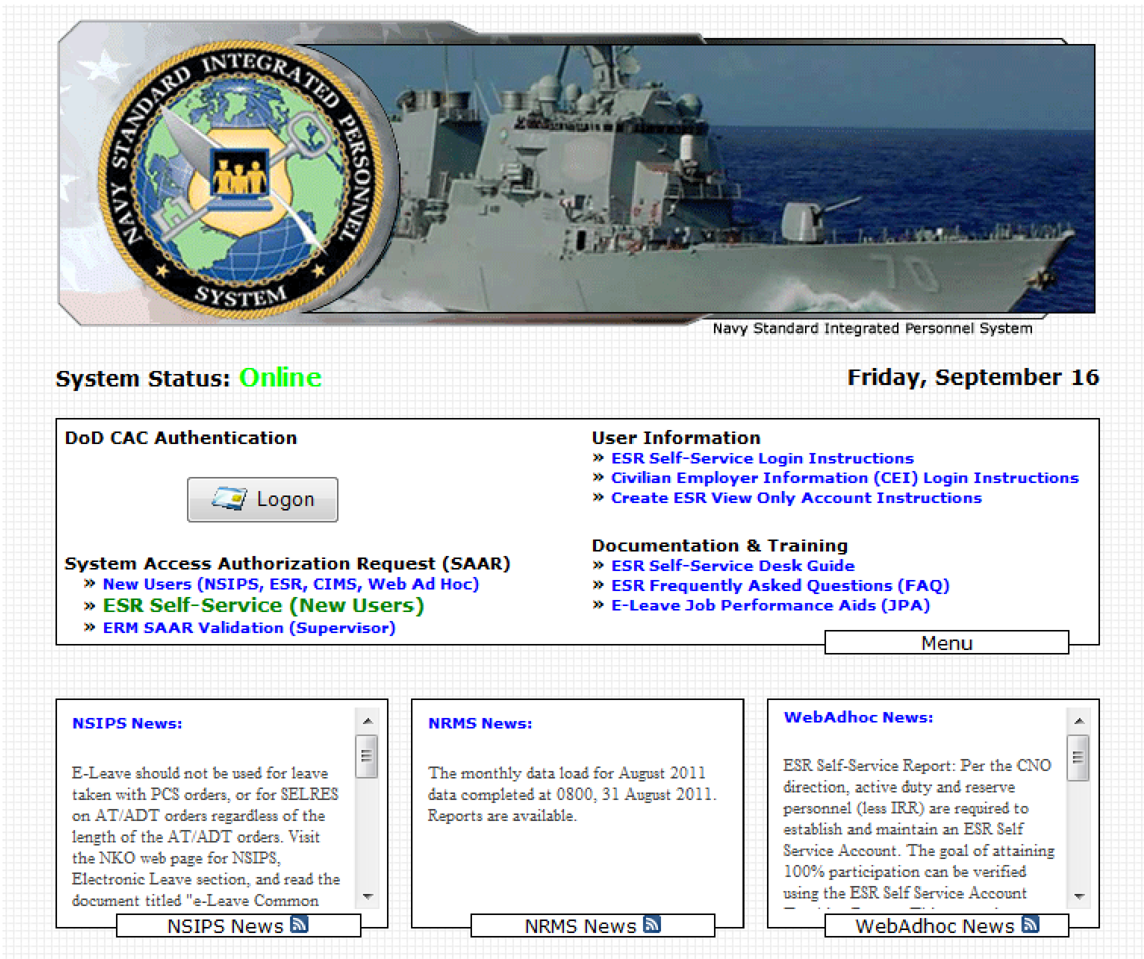


Figure 22. NSIPS/ESR Login Screen (From USN, n.d.)

Once the user has been authenticated and agreed to the terms of use, access of the ESR home page is granted. The home page allows users to click the view tab or task tab depending on how they intend to use the Web Application. The view tab allows users to open a read only version of the information associated with the links in Figure 23. The task tab allows the user to perform data entries as well as view select information under each of the links in Figure 23. NSIPS/ESR allows users the ability to administer and control a significant portion of their service records as well as automates the leave process.

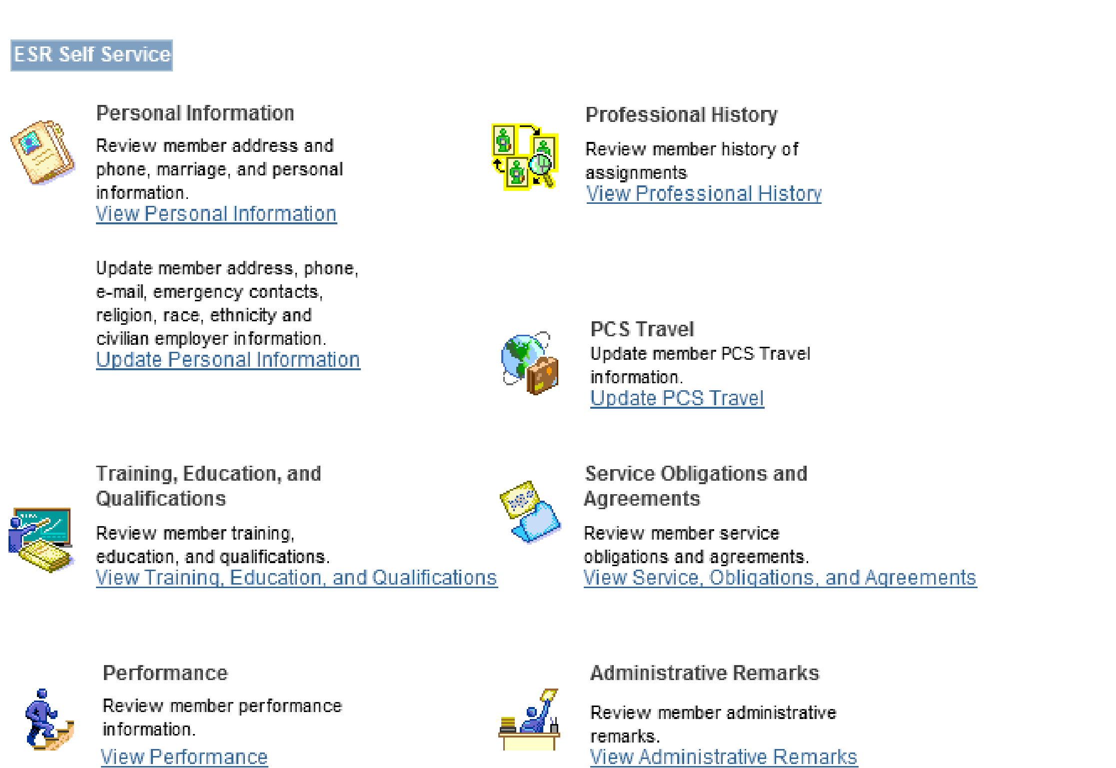


Figure 23. ESR Home Page (From USN, n.d.)

3. Defense Travel System (DTS)/Navy Reserve Order Writing System (NROWS)

NROWS is the DON Reservists enterprise wide Web Based Application used to manage the order writing process from start to finish for annual training, active duty training, and inactive duty training orders (SPAWAR, n.d.). This system automates the orders application process by integrating the approval process, budgeting, delivery of orders, and travel plans (SPAWAR, n.d.). This site may be accessed via the Navy Reserve Homeport Web Portal at http://www.navyreserve.navy.mil/CNRFC/pages. The link at the top labeled DTS/NROWS initiates a secure browser session requiring a valid CAC certificate and password for access. A validated user then gains access to the

DTS/NROWS Web Application. It should be noted that this Web Application requires the use of a Client capable of supporting a CAC reader and associated software. Figure 24 shows the Navy Reserve Home Portal and link for DTS/NROWS access via secure browser.

Figure 24. Navy Reserve Homeport Web Portal (From Navy Reserve, n.d.)

4. CNRFC Web Portal

The CNRFC Web Portal is a current initiative to provide access to a variety of Web Based Applications. The Web Portal is accessible from any device with Internet access and a web browser. The web address for accessing the CNRFC Web Portal is http://naval-reserve.com/. Clicking on any of the links initiates a secure browser session requiring username and password to authenticate. Providing the correct credentials allows the user to access that particular Web Based application. Figure 25 shows the CNRFC Web Portal and Web Application links. It is important to point out the links for Future Application 1 and Future Application 2. These should be populated as existing Web Based Applications are optimized and more Native Applications migrate to Web

Based solutions with the DON’s transition to NGEN. One such possible Web Based Application is Inactive Duty Training or (IDT) Order Writer.

Figure 25. CNRFC Web Portal (From CNRFC, 2011)

5. Navy Individual Augmentee (IA) Mobile Native Application

The IA Mobile Native Application was commissioned and developed by US Navy Fleet Forces Command for use with the iPhone and iPad. U.S. Navy Individual Augmentees that possess one of these devices and an iTunes account can download the Native Application to their Smart Mobile Wireless Device using iTunes or the App Store on the device itself. This Native Application provides the service member with up to date information and resources on the IA process (U.S. Fleet Forces Command, 2011). It allows the active duty service members family and parent command to keep abreast of the service member while deployed in an IA status (U.S. Fleet Forces Command, 2011). Reservist personnel preparing for an IA can use the application as mentioned, above as well as keep civilian employers up to date on their activities. The IA Native Application gives the user access to information on Navy Mobilization Processing Site (NMPS) and Inside the Continental United States (INCONUS) processing procedures from start to finish (U.S. Fleet Forces Command, 2011). The application provides access to IA related videos, Navy Publications, Navy Administrative Messages (NAVADMINS), and instructions (U.S. Fleet Forces Command, 2011). These are embedded in the downloaded app allowing access without an internet connection or wireless signal (U.S. Fleet Forces Command, 2011). Information and instructions on how to download the IA Native Application can be found at: http://itunes.apple.com/app/navy-ia/id329935269?mt=8. Figure 26 is a screen shot of the IA Mobile Native Application for the iPhone/iPad running iOS.

Figure 26. iPhone/iPad Mobile Native Application Screen Shots (From U.S. Fleet Forces Command, 2011)

C. STUDENT DEVELOPED WEB APPLICATION

The goal of this project was to develop a database and Web Accessible application to create a searchable Terrorist Incident Clearinghouse System in less than twelve weeks (one NPS quarter). LT Charles Fulmer and LT Jeremy Britt developed and built a prototype capable of retrieving information considered pertinent for reported terrorist incidents worldwide in support of this goal in about four weeks. The WITS Web Application can be accessed using any device with a web browser. Reported terrorist incidents from 2004 to 2010 were used to construct the database. The Worldwide Incidents Tracking System or WITS prototype was developed using a single physical machine acting as a web server, database server, and application server. The physical machine used consisted of a PowerEdge T310 Chassis allowing up to 4 Cabled Hard Drives. The OS used was Windows 2008R2 (x64) which includes Web Server IIS 7.5. The database server was developed using MySQL open source version 5.1. The application server used was Adobe ColdFusion version 9.1. The WITS Web Application can be accessed at the following Web address: http://suncoastsystem.com/wits/. The following sections will provide screen shots and explanations. The E-R diagram, Logical data model, database schema, and MySQL code are attached as the Appendix.

Figure 27 is a screen shot of the WITS Web Application and links to the questions answered using the MySQL queries. The screen shot also shows the pie graph of all incidents reported by country using the included legend. The Q links are for the specific questions addressed by the project objectives, but can be modified by selecting any of the countries via the scroll menu and clicking search as shown in Figure 28. This takes you to the view screen and displays all incidents by data, type, country of origin, city, dead count, wounded count, facility type, and damage. Figure 29 shows the returned data for Q1 modified for Afghanistan. Another example of note is LT Charles Fulmer's Web Based Search engine for DoD reports located at the following web address: http://dodreports.com/.

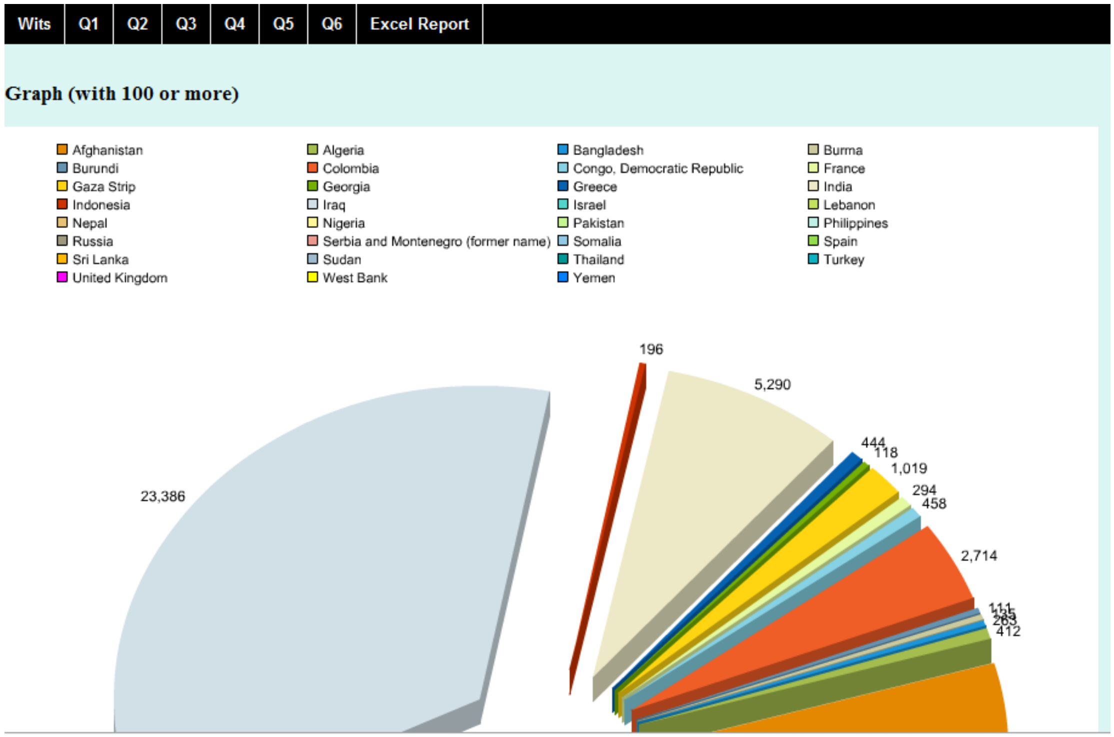


Figure 27. WITS Web Based Application Screen Shot

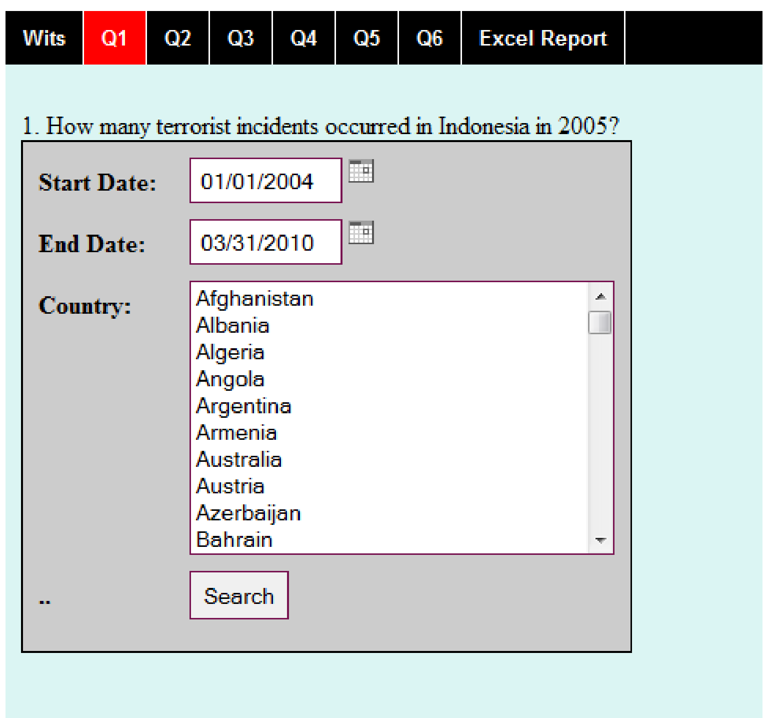


Figure 28. WITS Q1 Modifiable Search Screen Shot

Wits | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Excel Report

View

Incident Date	Event Type	Country	City	Dead Count	Wounded Count	Facility Type	Damage	Details	Save
01/04/2004	Bombing	Afghanistan	Kabul	0	0	Government	None	View	Save
01/05/2004	Bombing	Afghanistan	Kandahar	0	0	Humanitarian/NGO	Light	View	Save
01/07/2004	Armed Attack	Afghanistan		12	0	Bus	Light	View	Save
01/11/2004	Armed Attack	Afghanistan	Gerdi Sari	1	0	Vehicle	Light	View	Save
01/11/2004	Armed Attack	Afghanistan	Jalalabad	0	0	Military	None	View	Save
01/12/2004	Bombing	Afghanistan	Mazar-e Sharif	0	1	Humanitarian/NGO	Light	View	Save
01/14/2004	Armed Attack	Afghanistan	Khowst	10	0	Religious	Light	View	Save
01/19/2004	Armed Attack	Afghanistan	Eshashem	3	2	Police	Light	View	Save
01/19/2004	Bombing	Afghanistan	Kandahar	3	10	School/Educational	Light	View	Save
01/21/2004	Bombing	Afghanistan		5	1	Vehicle	Light	View	Save

Figure 29. WITS Incident Search Return Screen Shot

D. SUMMARY

Chapter IV applies concepts from the Chapter III TCO analysis to NMCI cost data provided by CNRFC and extrapolates to provide a cost comparison of current NMCI solutions vs. migration to Thin Clients. The migration to Thin Clients facilitates the need to continue the DON's migration of Native Applications to Web Based Applications. Chapter IV section B addresses the current initiatives and provides examples of their functionality and applicability for Smart Mobile Wireless Devices . This sections also analyzes and provides an example of a DON Native Application developed specifically for the iPhone/iPad and hosted by Apple. The final section shows the ease of development, accessibility, and adaptability of migrating to Web Based Applications in support of Thin Clients and future Smart Mobile Wireless Device initiatives. Chapter V will summarize the information in Chapters I through IV and provide conclusions for the reader. Finally, Chapter V will use those conclusions and the insights gained from this research to provide the reader with recommendations on future IT initiatives in support of the DON's objectives.

THIS PAGE INTENTIONALLY LEFT BLANK

V. CONCLUSIONS AND RECOMMENDATIONS

A. SUMMARY

This thesis researched and analyzed the advantages and disadvantages of the Thin Client-Server Architecture and Web Based Applications. The origin and evolution of applications and the technologies that supported them was presented to give the reader a thorough understanding of how current IT infrastructure came to fruition. Hardware and software developments have changed the IT landscape over time and required organizations to adjust to remain competitive and efficient. The developments in Thin Client technology, Web Based Applications, and Smart Mobile Wireless Devices were identified and researched to discover their viability as IT solutions for the DON. TCO was identified as an effective method of cost analysis to compare and contrast Fat and Thin Client costs. Energy Consumption was also identified as an effective means of a direct cost comparison using the average power consumed by a Fat Client vs. Thin Client solution. Fat Client-Server Architectures were analyzed and compared to the advantages vs. disadvantages of migrating to Thin Client-Server Architecture. The advantages and disadvantages of Native and Web Based Applications were analyzed to determine the best fit for the DON and how each might affect future initiatives. Cost and application data were provided by CNRFC for analysis and the results were extrapolated to allow for an organizational view. Existing successful DoD and DON Web Applications were identified and examples provided to show current functioning examples that may be emulated for future initiatives. Finally, a Web Based Application developed by students LT Chuck Fulmer and LT Jeremy Britt was included to show proof of concept.

The directives set forth by the DDCIO (N) require Echelon II Commands to reduce data centers by 25 percent, increase server utilization by at least 40 percent, and explore viable measures to increase IT efficiencies (Dorsett, 2011). The research in this thesis was conducted with these goals in mind. The decreasing cost of hardware has led to decentralization of IT infrastructure across the DoD. As the cost of hardware decreased it facilitated the exponential increase of software complexity and size. NMCI was created to standardize these IT services and eliminate as much of the decentralized Fat Client-Server architecture as possible. The end of the NMCI contract provides the DON an opportunity to take advantage of advancements in Thin Client-Server Architecture and Web Applications. The TCO of Thin Clients from the Gartner Research Group case study showed a cost savings of 32% over Fat Unmanaged Clients and 3.3% over Fat Managed Clients. The Power Consumption of the Fat Client with monitor tested was 170 watts vs. an average of 70 watts for the Thin Clients with monitor. The cost analysis shows that the DON can greatly reduce IT costs while supporting Navy Green IT efforts by migrating to a Thin Client solution.

The analysis of Thin Client Server Architecture revealed that the advantages of migrating to Thin Clients goes beyond TCO savings and Environmental efforts. Switching to Thin Client solutions facilitates meeting the DDCIO (N)'s data center reduction and server utilization objectives. One of the primary benefits of this architecture is the shift of processing to the server. The lack of processing capability inherent with using Thin Client devices increases server utilization significantly, while reducing the overall TCO. Software and application licensing are one of the largest costs associated with Fat Clients. Migrating to a Thin Client solution allows the Software and Applications to be moved to the server. The organization only pays for the server side licensing and the user access it needs. The migrating of applications and software to the server also decreases the software maintenance and management burden on IT staff, further reducing costs.

The use of Thin Client-Server Architecture will require the migration of Native Applications to Web Based Applications. There will be some costs associated with this as outlined in the TCO analysis. However, the migration costs are more than offset by the overall savings and increased life cycle of the Thin Client. The decrease in licensing fees also helps to offset migration costs. Web Based Applications have advanced significantly over the last few years and are now comparable to Native Applications in terms of functionality and user experience. The commercial trend toward centralized services and the expansion of Web Development Tools has made Web Based

Applications a viable alternative to Native Applications. CNRFC along with other DoD and DON organizations have already begun migrating to Web Based Applications due to their accessibility, flexibility, and straightforward administration. Power Users requiring significant processing power to run graphics intensive or multimedia applications may have to retain their Fat Client. This in no way excludes them from taking advantage of the Thin Client-Server architecture. There are numerous Thin Client solutions that allow Power Users to retain their existing Fat Client and use it in conjunction with a Thin Client to access shared services. Power Users generally make up a very small percentage of DON organizations. DON organizations that have a high volume of these users will need to seek alternative IT solutions.

Web Based Applications also allow for expansion of future initiatives. Smart Mobile Wireless Devices continue to proliferate in the private and commercial sectors. Commands like CNRFC interface with Reservists on all levels and require a solution that allows the organization to take advantage of these devices with maximum flexibility and minimum cost. Web Based Applications can be optimized for device specific operating systems using software development kits. The same advantages that Web Based Applications offer Thin Clients apply to Mobile Web Applications. The largest advantage Mobile Web Applications have over Mobile Native Applications is the cost of development and administration. Mobile Web Applications can be standardized across a broad range of Mobile Operating Systems and Devices. Mobile Native Applications must be written using the Native Code unique to each device's OS. The minute increases in user experience or usability do not justify the increase in cost or development time in most cases. Only personnel requiring the ability to access application data with no Internet or Cellular access have a justifiable need for the development of a Mobile Native Application. One example is the IA Native Application for the iPhone presented in Chapter IV. The primary issue for IA personnel is lack of connectivity, which may justify the use of the Mobile Native Application. However, only those personnel possessing an iPhone or iPad have access to this application. To make this Mobile Native

Application available to all Smart Mobile Wireless Device users would require extensive development and administration costs. It is up to the command to determine if the increased functionality is worth the cost.

The TCO case study was used as a cost model to analyze the CNRFC data. The average savings of migrating to a Thin Client solution was 17.65%. This assumes that the IT service provider for NGEN passes on the cost savings. If we negate the Thin Client savings, CNRFC could still achieve an average energy cost savings of 37.3% using the data provided. Base on the research and analysis conducted in this thesis, CNRFC and similar DON shore commands would benefit from migrating to a Thin Client solution. CNRFC and other DON shore commands have already initiated migrating some Native Applications to Web Based Applications. The advantages of Web Based Applications over Native Applications identified in this thesis supports these and future initiatives as they transition to NGEN. Finally, Web Applications facilitate the migration to Thin Client-Server Architecture and the ability to incorporate Smart Mobile Wireless technologies as the DON evolves to meet new challenges.

B. AREAS FOR FURTHER STUDY

This thesis focused on shore based Naval Installations and unclassified systems. The author recommends examining the feasibility of using Thin Clients with classified systems. Further study might also include the examination of how ship systems will interface with shore commands incorporating Thin Client-Server Architectures. Another key area for further study is the incorporation of wireless connectivity at shore commands allowing Intranet access to Smart Mobile Wireless Device users. Shore Commands supporting Reservists personnel would be the most likely candidates to benefit from Wi-Fi. Areas considered outside the scope of this thesis are: Network Security, Smart Mobile Wireless Device Security, CAC authentication issues, User Bias, and User Resistance to Change.

APPENDIX

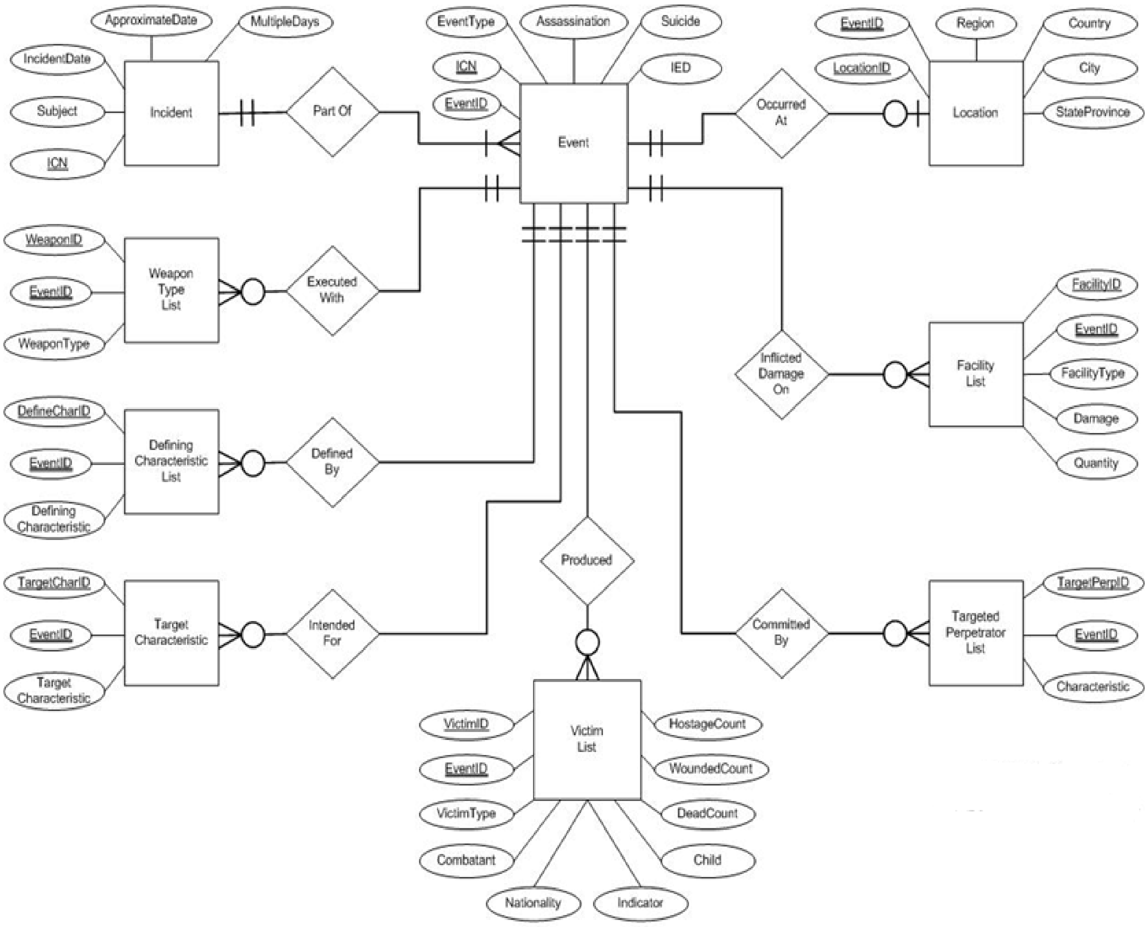


Figure 30. WITS Database Entity Relationship Diagram

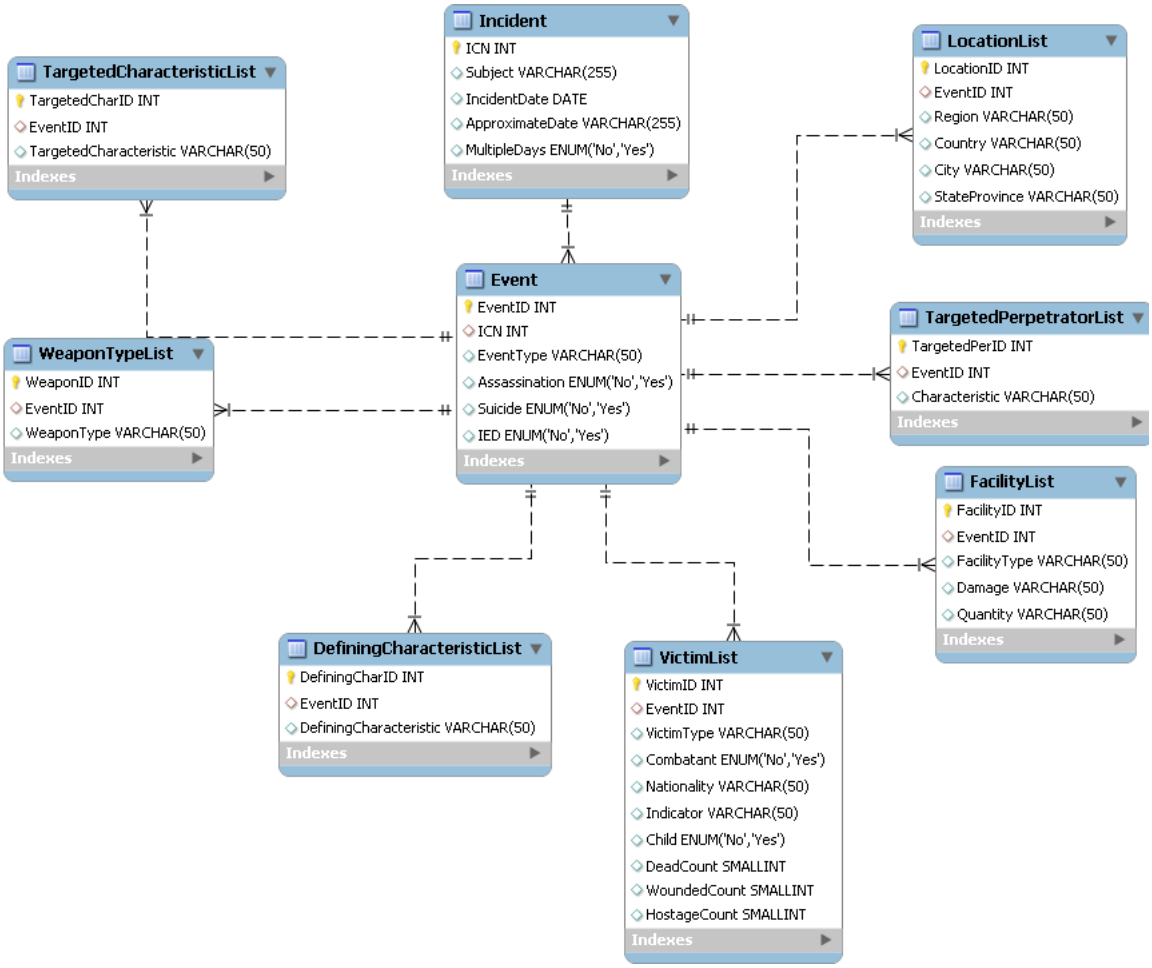


Figure 31. WITS Database Logical Data Model

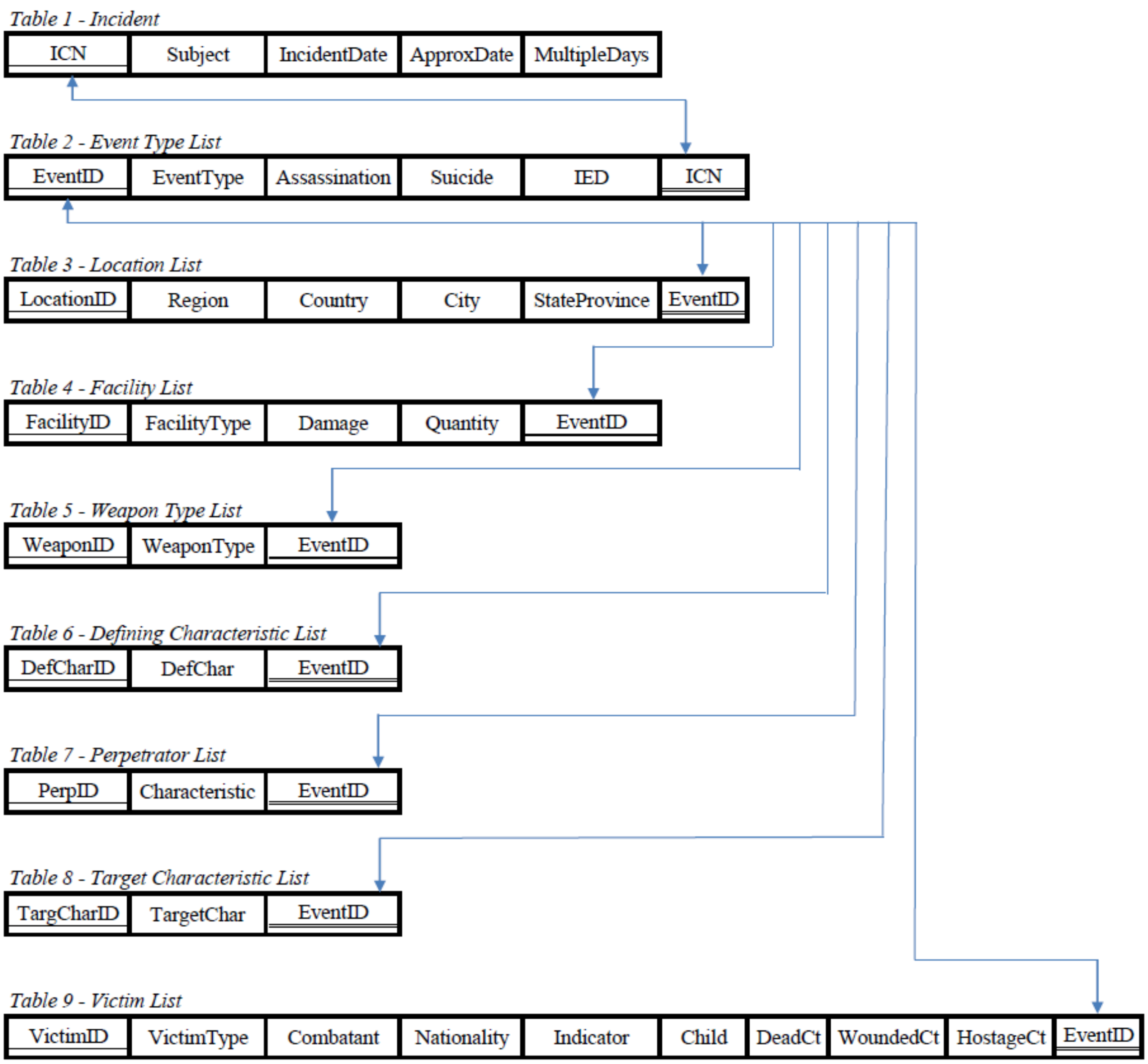


Figure 32. WITS Database System

Q1. How many terrorist incidents occurred in Indonesia in 2005?

```
SELECT incident.Subject, incident.IncidentDate, locationlist.Country
FROM incident
INNER JOIN event ON incident.ICN = event.ICN
INNER JOIN locationlist ON event.EventID = locationlist.EventID
WHERE locationlist.Country =  'Indonesia' AND incident.IncidentDate
BETWEEN  '2005-01-01' AND '2005-12-31'
```

Q2. How many improvised explosive devises (IED) attacks have been prosecuted in Iraq and Afghanistan?

```
SELECT incident.Subject, locationlist.Country, event.IED
FROM incident
INNER JOIN event ON incident.ICN = event.ICN
INNER JOIN locationlist ON event.EventID = locationlist.EventID
WHERE locationlist.Country IN ('Iraq','Afghanistan') AND event.ied = 'Yes'
```

Q3. Are hostage incidents on the rise in Chechnya?

```
SELECT locationlist.StateProvince, incident.Subject, locationlist.StateProvince,
Sum(victimlist.HostageCount) AS HostageCount,
Year(incident.IncidentDate) AS IncidentYear,
Count(event.EventID) AS IncidentCount
FROM incident
INNER JOIN event ON incident.ICN = event.ICN
INNER JOIN locationlist ON event.EventID = locationlist.EventID
INNER JOIN victimlist ON event.EventID = victimlist.VictimID
WHERE locationlist.StateProvince IN ('Chechnya') AND HostageCount > 0
GROUP BY IncidentYear
```

Q4. About how far apart are fatal attacks occurring in the Philippines?

```
DROP TEMPORARY TABLE IF EXISTS dcheckA;
DROP TEMPORARY TABLE IF EXISTS dcheckB;

CREATE TEMPORARY TABLE dcheckA (
id int NOT NULL auto_increment,
dt date,
PRIMARY KEY (id));
CREATE TEMPORARY TABLE dcheckB (
id int NOT NULL auto_increment,
dt2 date,
PRIMARY KEY (id));
INSERT INTO dcheckA (dt)
SELECT DATE(incident.IncidentDate) AS IncidentDate
FROM event
Inner Join incident ON incident.ICN = event.ICN
Inner Join locationlist ON event.EventID = locationlist.EventID
WHERE locationlist.Country = 'Philippines' AND  incident.IncidentDate BETWEEN '2009-05-30' AND '2009-06-29'
ORDER BY incident.IncidentDate
LIMIT 0, 23;
INSERT INTO dcheckB (dt2)
SELECT DATE(incident.IncidentDate) AS IncidentDate
FROM event
Inner Join incident ON incident.ICN = event.ICN
Inner Join locationlist ON event.EventID = locationlist.EventID
WHERE locationlist.Country = 'Philippines' AND  incident.IncidentDate BETWEEN '2009-05-30' AND '2009-06-29'
ORDER BY incident.IncidentDate
LIMIT 1, 23;

SELECT dcheckA.dt AS 'Start Date', dcheckB.dt2 AS 'End Date', DATEDIFF(dcheckB.dt2, dcheckA.dt) AS 'Time Interval'
FROM dcheckA, dcheckB
WHERE dcheckA.id = dcheckB.id;
```

Q5. In what years were the Revolutionary Armed Forces of Columbia (FARC) most active?

```
SELECT locationlist.StateProvince, incident.Subject, locationlist.StateProvince,
Sum(victimlist.HostageCount) AS HostageCount,
Year(incident.IncidentDate) AS IncidentYear,
Count(event.EventID) IncidentCount
FROM incident
INNER JOIN event ON incident.ICN = event.ICN
INNER JOIN locationlist ON event.EventID = locationlist.EventID
INNER JOIN victimlist ON event.EventID = victimlist.VictimID
WHERE incident.Subject LIKE '%FARC%'
GROUP BY IncidentYear
```

Q6. How often are small arms used in attacks in Central Asia?

```
SELECT locationlist.Country, weapontypelist.WeaponType
FROM incident
INNER JOIN event ON incident.ICN = event.ICN
INNER JOIN locationlist ON event.EventID = locationlist.EventID
INNER JOIN weapontypelist ON event.EventID = weapontypelist.EventID
WHERE locationlist.Country IN ('Afghanistan', 'Kazakhstan', 'Kyrgyzstan',
'Tajikistan', 'Uzbekistan') AND weapontypelist.WeaponType IN ('Firearm')
```

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

Anderson, C., & Greenberg, S. (n.d.). Desktop energy consumption. *Wyse.com*. Retrieved August 25, 2011, from http://www.wyse.com/resources/whitepapers/energy_study.pdf

Becta. (2004, September). Thin client. *Scribd*. Retrieved September 21, 2011, from http://www.scribd.com/doc/28851651/Thin-Client

Best Buy–Computers, Video Games, TVs, Cameras, Appliances, Phones. (2011). Retrieved September 21, 2011, from http://www.bestbuy.com/

Ceruzzi, P. E. (2003). *History of modern computing (2nd edition)*. Cambridge, MA, USA: MIT Press.

Charland, A., & LeRoux, B. (2011). Mobile application development: Web vs. native. *Queue, 9*(4), 20:20-20:28. doi:http://doi.acm.org/10.1145/1966989.1968203

"Client Server," *Encyclopedia of Computer Science* (2003). In Ralston A., Reilly E. D. and Hemmendinger D. (Eds.), (4th ed.). Chichester, UK: John Wiley and Sons Ltd.

Clouse, S. (n.d.). Web-based applications vs. client-server software. *Fcny.org*. Retrieved August 29, 2011, from http://metrix.fcny.org/wiki/download/attachments/7328/Web-based+Apps+vs+Client-Server+Software.pdf?version=1

CNRFC. (2011). *Commander, Naval Reserve Forces Command*. Retrieved September 16, 2011, from http://naval-reserve.com/ Daniel, S. E. (1996). Client/server computing: Applications & issues. *Journal of Systems Management, 47*(3), 58-58. Retrieved from http://search.proquest.com/docview/199820572?accountid=12702

David, B. (2002, March 27). Thin client benefits. *Thinclient.net*. Retrieved August 14, 2011, from http://www.thinclient.net/pdf/Thin_Client_Benefits_Paper.pdf

Dean, T. (2006). *Network+ guide to networks* (4th ed.). Boston: Thomson/Course Technology.

DELL. (2011). Documentation. *Welcome to Dell technical support*. Retrieved September 15, 2011, from http://support.dell.com/support/edocs/monitors/E1X0S/en/ug/about.htm#Specifications

DELL. (n.d.). Technical guidebook inside the Optiplex 380. *Dell.com*. Retrieved September 15, 2011, from http://i.dell.com/sites/content/business/solutions/engineering-docs/en/Documents/optiplex-380-technical-guidebook.pdf

Device APIs Working Group - W3C. (2011, September 15). *World wide web consortium (W3C)*. Retrieved August 31, 2011, from http://www.w3.org/2009/dap/

DFAS. (n.d.). MyPay. *MyPay mobile*. Retrieved September 17, 2011, from https://mypay.dfas.mil/FAQ.htm#myPayMobile

DFAS. (n.d.). MyPay Web Site. *MyPay web site*. Retrieved September 16, 2011, from https://mypay.dfas.mil/mypay.aspx

Dorsett, D. J. (2011, January 13). NAVADMIN 008/11. *U.S. Navy hosting*. Retrieved September 3, 2011, from http://www.public.navy.mil/bupers-npc/reference/messages/Documents/NAVADMINS/NAV2011/NAV11008.txt

Electric Power Monthly. (2011, September 15). *U.S. Energy Information Administration (EIA)*. Retrieved September 9, 2011, from http://www.eia.gov/electricity/monthly/index.cfm

Goldstine, H. H., & Goldstine, A. (1996). The electronic numerical integrator and computer (ENIAC). *IEEE Annals of the History of Computing, 18*(1), 10-10-16. doi:DE6B2A1F-8E7A-4D4B-B65F-BA9AFCC154DB

Heydler/Wyse Federal Insides Sales Rep, K. (2011, September 14). Wyse Thin Client Product Information [E-mail to the author].

Lowber, P. (2001, September 28). Thin-client vs. fat-client TCO. *Hp.com*. Retrieved August 24, 2011, from http://h20219.www2.hp.com/Hpsub/downloads/WhitePaper_Gartner_ThinClient_TCO.pdf

Machovec, G. S. (1997, January). Thin client architecture: The promise and the problems. *Naval Postgraduate School - Academics - Home*. Retrieved September 21, 2011, from http://search.proquest.com.libproxy.nps.edu/docview/206816370/fulltextPDF?accountid=12702

Mallick, M. (2003). *Mobile and wireless design essentials*. Indianapolis, IN: Wiley.

“Mobile.” *Dictionary and thesaurus - Merriam-Webster online* Retrieved 8/18/2011, 2011, from http://www.merriam-webster.com/

Moss/CNRFC N63, B. D. (2011, September 9). NMCI CLIN Data [E-mail to the author].

Navy Reserve. (n.d.). *Navy.mil*. Retrieved September 17, 2011, from https://www.navyreserve.navy.mil/Pages/default.aspx

O'Regan, G. (2008). *A brief history of computing*. London: Springer. Retrieved from http://libproxy.nps.edu/login?url=http://www.springerlink.com/content/h06437/?p=fb8049804af1437fa8b538aec4bf0bda&pi=0

PDAs & Smartphones - Wireless From AT&T Retrieved 8/19/2011, 2011, from http://www.wireless.att.com/cell-phone-service/cell-phones/pda-phones-smartphones.jsp#sku5100284_a|||

Phifer, L. (2000, August). What is 3G (third generation of mobile telephony)? - Definition from Whatis.com. *Telecom Information, News and Tips - SearchTelecom.com*. Retrieved September 21, 2011, from http://searchtelecom.techtarget.com/definition/3G

Ponniah, P. (2005, January 28). Database design and development: An essential guide for IT professionals - Ponniah. *Wiley Online Library*. doi: 10.1002/0471728993

Research In Motion. (2009). *RIM Company - Learn about Research In Motion*. Retrieved September 21, 2011, from http://www.rim.net/products/handhelds/specifications/index.shtml

Samsung Convoy | Samsung SCH-u640 - Cell Phones. (n.d.). *SAMSUNG*. Retrieved September 21, 2011, from http://www.samsung.com/us/mobile/cell-phones/SCH-U640EKAVZW

Sintes, JavaWorld.com, 08/23/02, B. (n.d.). App server, Web server: What's the difference? - JavaWorld. *Welcome to JavaWorld.com*. Retrieved August 8, 2011, from http://www.javaworld.com/javaworld/javaqa/2002-08/01-qa-0823-appvswebserver.html

Schmidt, M. J. (2011, September 4). Total Cost of Ownership (TCO) defined, explained, and calculated. *Business Case Books, Tools, and Professional Developrment Training*. Retrieved August 24, 2011, from http://www.solutionmatrix.com/total-cost-of-ownership.html

"Smart." *Dictionary and thesaurus - Merriam-Webster online* Retrieved 8/18/2011, 2011, from http://www.merriam-webster.com/

SPAWAR. (n.d.). Navy Reserve Systems. *U.S. Navy Hosting*. Retrieved September 16, 2011, from http://www.public.navy.mil/spawar/Atlantic/ProductsServices/Pages/NavyReserveSystems.aspx

Sternstein, A. (2010, July 9). Navy awards contract to transition to expanded network - Nextgov. *Nextgov - Federal Technology News, Best Practices, and Web 2.0 Tools*. Retrieved September 12, 2011, from http://www.nextgov.com/nextgov/ng_20100709_9862.php?oref=topnews

Sun Ray virtual display clients. (2011, February 28). *Sun Ray User Group*. Retrieved August 13, 2011, from http://www.sun-rays.org/clients.html

USN. (n.d.). *NSIPS ESR*. Retrieved September 15, 2011, from https://nsips.nmci.navy.mil/

U.S. Fleet Forces Command. (2011). Navy IA. *App Store - Navy IA*. Retrieved September 17, 2011, from http://itunes.apple.com/app/navy-ia/id329935269?mt=8

Urbanowicz, J. (2001). Client-server architecture: What's the ideal strategy? *Unisys World*, *22*(5), 1-1,14+. Retrieved from http://search.proquest.com/docview/274454118?accountid=12702

"Wireless." *Dictionary and thesaurus - Merriam-Webster online* Retrieved 8/18/2011, 2011, from http://www.merriam-webster.com/

Wyse Technology - Wyse C90LEW thin client. (n.d.). *Wyse Technology - The Global Leader in Cloud Client Computing*. Retrieved August 21, 2011, from http://www.wyse.com/products/hardware/thinclients/C90LEW/index.asp

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Prof. Dan Boger
   Chair, Department of Information Sciences
   Naval Postgraduate School
   Monterey, California

4. Prof. Glenn R. Cook
   Department of Information Sciences, Cod IS/Pr
   Naval Postgraduate School
   Monterey, California

5. Prof. Doug MacKinnon
   Department of Information Sciences, Cod IS/Pr
   Naval Postgraduate School
   Monterey, California

6. LT Jeremy L. Britt
   Birmingham, Alabama